

Motivation

Dienstag, 21. Januar 2020 11:18

Gegeben eine Theorie Γ und einen Satz Φ , bestimme, ob
 $\Gamma \models \Phi$ gilt.

Aufgabe: Finde eine möglichst minimale Teilmenge

$\Gamma^* \subseteq \Gamma$ so dass $\Gamma \models \Phi \Leftrightarrow \Gamma^* \models \Phi$ gilt.

Implementiere dazu eine Selektionsfunktion $\sigma(\Gamma) = \Gamma^*$.

Idee: Vector Space Model mit

$$\sigma(\Gamma) = \{ \vec{\gamma} \in \Gamma \mid \|\vec{\Phi} - \vec{\gamma}\| \leq \delta \},$$

wobei $-\vec{\Phi}$ und $\vec{\gamma}$ aus einem geeigneten Vektorraum stammen

- $\|\cdot\|$ eine geeignete Norm ist und
- δ ein geeigneter Schwellwert ist.

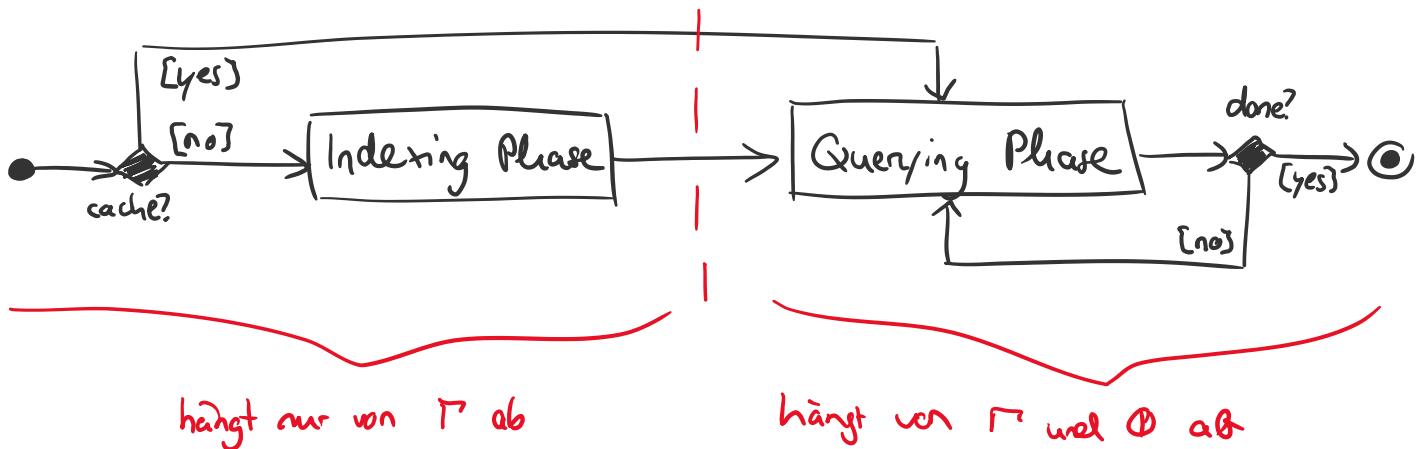
Algorithmus

Dienstag, 21. Januar 2020

11:33

- Libraries:
- gavell/Eprover für Parsing & Proving
 - gensim für Modeling & Indexing

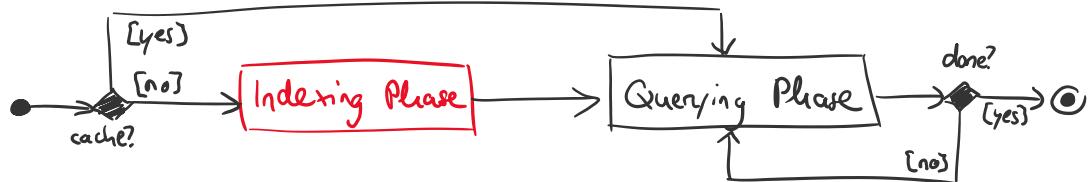
Der Algorithmus besteht aus 2 Phasen:



Indexing Phase

Dienstag, 21. Januar 2020

11:45



1. Axiome vektorisieren

- Ein Axiom ist eine (Multi-)Menge von Symbolen t_i . (s. S. NE)
- Zähle Vorkommen von Symbolen und bau ein passende (Sparse-)Vektoren.

Beispiel:

$$\begin{aligned} \exists y: & \text{has-part}(a\text{-cell}, y) \\ \rightsquigarrow & \{\text{has-part}, a\text{-cell}\} \\ \rightsquigarrow & \begin{pmatrix} \text{has-part} & a\text{-cell} & \dots & \dots & \dots \\ 1 & 1 & 0 & \dots & 0 \end{pmatrix} \end{aligned}$$

$$\begin{array}{c|c} \gamma_1 & \\ \gamma_2 & \\ \vdots & \\ \gamma_n & \\ \hline \emptyset & \end{array} \rightsquigarrow \begin{matrix} t_1 & & & \\ t_2 & & & \\ \vdots & & & \\ t_m & & & \end{matrix} \begin{pmatrix} | & | & | \\ \overrightarrow{\gamma_1} & \overrightarrow{\gamma_2} & \dots & \overrightarrow{\gamma_n} \\ | & | & & | \end{pmatrix}$$

Term - Document - Matrix A
(oder: Symbol - Axiom - Matrix)

2. TF-IDF-Modell anwenden

- bisher zählen wir Symbolvorkommen (term frequencies):

$$A_{ij} = tf_{ij} \leftarrow \text{wie oft kommt } t_i \text{ in } \gamma_j \text{ vor?}$$

- berücksichtige auch die Seltenheit von Symbolen, um "common symbols" zu behandeln.

- Seltenheit (inverse document frequency):

$$idf_i = \log_2 \frac{|T|}{df_i} \leftarrow \begin{array}{l} \text{Anzahl Axiome} \\ \text{in wie vielen Axiomen kommt } t_i \text{ vor?} \end{array}$$

- berechne neue Matrix: $A'_{ij} = tf_{ij} \cdot idf_i$

1 2 3

1 2 3

- Berechne neue Matrix: $A'_{ij} = tf_{ij} \cdot idf_i$

Beispiel:

$$\textcircled{1} \quad T \Rightarrow a=b$$

$$\textcircled{2} \quad T \Rightarrow p(a,c) \rightsquigarrow A =$$

$$\textcircled{3} \quad p(b,c) \Rightarrow \perp$$

$$\therefore \perp$$

$$A = \begin{pmatrix} & \textcircled{1} & \textcircled{2} & \textcircled{3} \\ a & 1 & 1 & 0 \\ b & 1 & 0 & 1 \\ c & 0 & 1 & 1 \\ p & 1 & 0 & 0 \\ T & 1 & 1 & 0 \\ \perp & 0 & 0 & 1 \end{pmatrix}$$

$$idf_{a,b,c,p,T} \approx .6$$

$$idf_{\perp,1} \approx 1.6$$

$$A' = \begin{pmatrix} & \textcircled{1} & \textcircled{2} & \textcircled{3} \\ a & .6 & .6 & 0 \\ b & .6 & 0 & .6 \\ c & 0 & .6 & .6 \\ p & 1.6 & 0 & 0 \\ T & 0 & .6 & .6 \\ \perp & 0 & 0 & 1.6 \end{pmatrix}$$

3.1 LSI-Modell anwenden

Latent Semantic Indexing ist ein Verfahren des Information Retrieval.

Verfahren wie das **LSI** sind insbesondere für die **Suche auf großen Datenmengen** von Interesse.
Das Ziel von **LSI** ist es, **Hauptkomponenten von Dokumenten** zu finden.

Diese Hauptkomponenten (Konzepte) kann man sich als **generelle Begriffe** vorstellen.

Dieses Verfahren ist zum Beispiel dazu geeignet, aus sehr vielen Dokumenten diejenigen herauszufinden, die sich thematisch mit 'Autos' befassen, auch wenn in ihnen das Wort *Auto* nicht explizit vorkommt. Des Weiteren kann **LSI** dabei helfen, Artikel, in denen es wirklich um Autos geht, von denen zu unterscheiden, in denen nur das Wort *Auto* erwähnt wird.

Aus https://de.wikipedia.org/wiki/Latent_Semantic_Analysis

- Singulärwertzerlegung berechnen:

$$A' = U \cdot \Sigma \cdot V^T$$

term-topic matrix topic relevance topic-document matrix

$$\begin{aligned} & \left(\begin{array}{c|cccccc} C & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{cosmonaut} & 1 & 0 & 1 & 0 & 0 & 0 \\ \text{astronaut} & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{moon} & 1 & 1 & 0 & 0 & 0 & 0 \\ \text{car} & 1 & 0 & 0 & 1 & 1 & 0 \\ \text{truck} & 0 & 0 & 0 & 1 & 0 & 1 \end{array} \right) = U \Sigma V^T \\ & \text{sm=0} \\ & = \left(\begin{array}{c|cccccc} U & \text{Dim. 1} & \text{Dim. 2} & \text{Dim. 3} & \text{Dim. 4} & \text{Dim. 5} \\ \hline \text{cosmonaut} & -0.44 & -0.30 & 0.57 & 0.58 & 0.25 \\ \text{astronaut} & -0.13 & -0.33 & -0.59 & 0.00 & 0.73 \\ \text{moon} & -0.48 & -0.51 & -0.37 & 0.00 & -0.61 \\ \text{car} & -0.70 & 0.35 & 0.15 & -0.58 & 0.16 \\ \text{truck} & -0.26 & 0.65 & -0.41 & 0.58 & -0.09 \end{array} \right) \times \left(\begin{array}{cccccc} \Sigma & 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ & 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ & 0.00 & 0.00 & 1.28 & 0.00 & 0.00 \\ & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ & 0.00 & 0.00 & 0.00 & 0.00 & 0.39 \end{array} \right) \times \\ & \left(\begin{array}{c|cccccc} V^T & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{Dimension 1} & -0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\ \text{Dimension 2} & -0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\ \text{Dimension 3} & 0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\ \text{Dimension 4} & 0.00 & 0.00 & 0.58 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & -0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22 \end{array} \right) \end{aligned}$$

- Mit **Caw-Rank-Approximation** die Anzahl Dimensionen reduzieren:

$$C_k = U \cdot \Sigma_k \cdot V^T$$

($k =$ Anzahl gewünschter Dimensionen)

- A^k ist dann der **Ausschnitt** aus V^T , eine **topic-document-matrix**.

$$\left(\begin{array}{c|cccccc} C_2 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{cosmonaut} & 0.85 & 0.52 & 0.28 & 0.13 & 0.21 & -0.08 \\ \text{astronaut} & 0.36 & 0.36 & 0.16 & -0.20 & -0.02 & -0.18 \\ \text{moon} & 1.01 & 0.72 & 0.36 & -0.04 & 0.16 & -0.21 \\ \text{car} & 0.97 & 0.12 & 0.20 & 1.03 & 0.62 & 0.41 \\ \text{truck} & 0.12 & -0.39 & -0.08 & 0.90 & 0.41 & 0.49 \end{array} \right) = U \Sigma_2 V^T$$

$$\begin{aligned} & \left(\begin{array}{c|cccccc} U & \text{Dim. 1} & \text{Dim. 2} & \text{Dim. 3} & \text{Dim. 4} & \text{Dim. 5} \\ \hline \text{cosmonaut} & -0.44 & -0.30 & 0.57 & 0.58 & 0.25 \\ \text{astronaut} & -0.13 & -0.33 & -0.59 & 0.00 & 0.73 \\ \text{moon} & -0.48 & -0.51 & -0.37 & 0.00 & -0.61 \\ \text{car} & -0.70 & 0.35 & -0.15 & -0.58 & 0.16 \\ \text{truck} & -0.26 & 0.65 & -0.41 & 0.58 & -0.09 \end{array} \right) \times \left(\begin{array}{cccccc} \Sigma_2 & 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ & 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{array} \right) \times \\ & \left(\begin{array}{c|cccccc} V^T & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{Dimension 1} & -0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\ \text{Dimension 2} & -0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\ \text{Dimension 3} & 0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\ \text{Dimension 4} & 0.00 & 0.00 & 0.58 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & -0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22 \end{array} \right) \end{aligned}$$

Beispiel:

$$\textcircled{1} \quad T \Rightarrow a=b$$

$$\textcircled{2} \quad T \Rightarrow p(a,c) \rightsquigarrow A' =$$

$$\textcircled{3} \quad p(b,c) \Rightarrow \perp$$

$$A' = \begin{pmatrix} & \textcircled{1} & \textcircled{2} & \textcircled{3} \\ T & .6 & .6 & 0 \\ a & .6 & .6 & 0 \\ b & .6 & 0 & .6 \\ c & 0 & .6 & .6 \\ p & 0 & .6 & .6 \\ \perp & 0 & 0 & 1.6 \end{pmatrix}$$

$$\begin{aligned} & \begin{array}{ccc} \text{topic 1} & \text{topic 2} & \text{topic 3} \end{array} \\ & \begin{bmatrix} 0.52058344 & -0.3611615 & 0.01872013 \\ 0.52058344 & -0.3611615 & 0.01872013 \\ 0.37091602 & 0.02612715 & 0.68468126 \\ 0.35081748 & 0.20855771 & -0.48496723 \\ 0.35081748 & 0.20855771 & -0.48496723 \\ 0.27250918 & 0.80716478 & 0.2452025 \end{bmatrix} \begin{array}{c} T \\ a \\ b \\ c \\ p \\ \perp \end{array} \\ & \Sigma = \begin{bmatrix} 1.31951253 & 0.92400295 & 0.63647878 \end{bmatrix} \end{aligned}$$

$$\textcircled{3} \quad \frac{p(b,c) = \perp}{\therefore \perp}$$

$$P \begin{pmatrix} 0 & 0.6 & 0 \\ 0 & 0 & 1.6 \end{pmatrix}$$

$$\Sigma = \begin{bmatrix} 0.27250918 & 0.80716478 & 0.2452025 \end{bmatrix} \quad \perp$$

$$A'' = V^T = \begin{bmatrix} \textcircled{1} & \textcircled{2} & \textcircled{3} \\ \begin{bmatrix} 0.61785425 & 0.66039607 & 0.42677053 \\ -0.43495154 & -0.165106 & 0.88518763 \\ 0.65503686 & -0.73254149 & 0.18522884 \end{bmatrix} & \begin{matrix} \text{topic 1} \\ \text{topic 2} \\ \text{topic 3} \end{matrix} \end{bmatrix}$$

These: A'' ist kleiner als A' , was die Querying-Phase beschleunigt, und repräsentiert die "Themen" in der Theorie besser.

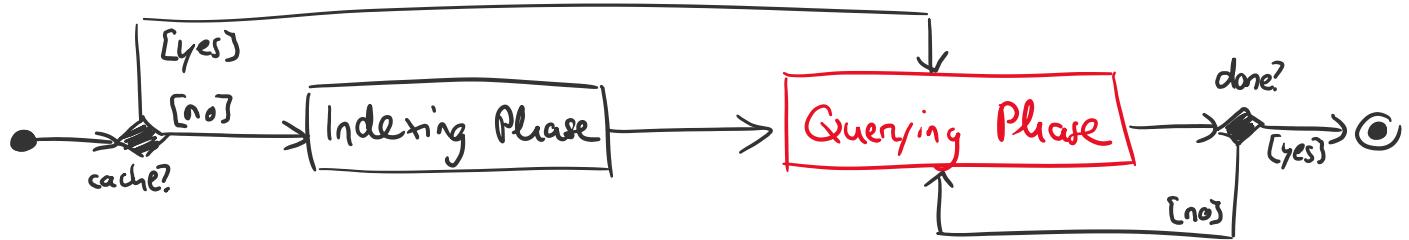
4.1 Index aufbauen

- Wir wollen gleiche Dokumente (= Axiome) finden, die einem gegebenen Dokument (= Conjecture) ähneln.
- Da A'' sehr groß werden kann, sollte man dafür effiziente Datenstrukturen verwenden.
- gensim kann das vollautomatisch.

In besondere kann der Index für andere Conjectures wieder verwendet werden.

Querying Phase

Dienstag, 21. Januar 2020 13:44



- Gegeben eine Conjecture Φ , finde "passende" Axiome Γ^* .
- Wann ist ein Axiom "passend"?

$$\Rightarrow \text{sim}(\vec{\gamma}_1, \vec{\gamma}_2) = \frac{\vec{\gamma}_1 \cdot \vec{\gamma}_2}{\|\vec{\gamma}_1\| \|\vec{\gamma}_2\|} \Rightarrow \text{cosine similarity maximieren.}$$

Algorithmus:

- Transformiere Φ in den LSI-Raum von vorhin ($\vec{\Phi}$).
- Für alle $\gamma \in \Gamma$, berechne $\text{sim}(\vec{\Phi}, \vec{\gamma})$, sortiere und nimmt die besten Ergebnisse als Γ^* .

```

def query_index(self, index: Similarity,
    premises: Iterable[Sentence], query: Sentence) -> Iterable[Sentence]:
    """Queries an index for the premises that match a given formula best."""

    with Message(f"Querying index for formula {query.name}"):
        similarities = index[transform_into_lsi_space(query)]

    with Message("Selecting best-matching axioms", show_done=not get_verbose()):
        return do_to(
            similarities, # take the similarity scores, then
            enumerate, # pair scores with premise indices
            partial(sorted, key=lambda e: -e[1]), # sort by descending similarity score
            partial(takewhile, lambda e: not self.select_until(...)), # select premises until done
            partial(map, lambda e: (premises[e[0]], e[1])), # map premise index to premise
            partial(map, first), # discard scores
            list # consume generator
        )

```

`select_until` ist ein geeignetes Abbruchkriterium.

Selektionsalgorithmus (naiv):

```

def select(self, problem: Problem) -> Problem:
    index = self.build_index(problem.premises) # indexing phase
    reduced_premises = self.query_index(index, problem.premises, query=problem.conjecture) # querying phase
    return Problem(premises=reduced_premises, conjecture=problem.conjecture)

```

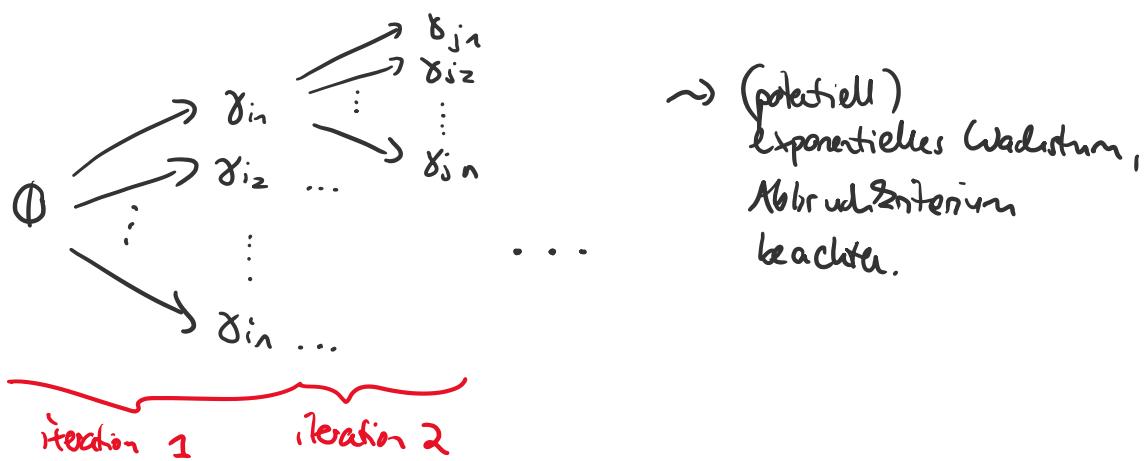
⇒ kam nur schwer Transitivität wie in $a=b \wedge p(a) \Rightarrow p(b)$
 erkennen (nur über S1).

Selektionsalgorithmus (iterativ):

```
def select(self, problem: Problem) -> Problem:
    index = self.build_index(problem.premises) # indexing phase

    # querying phase (a kind of "breadth-first search")
    reduced_premises = set()
    step = [problem.conjecture]
    for i in range(0, self.iterations):
        step = set([new_formula for formula in step for new_formula in
                   self.query_index(index, problem.premises, query=formula)])
        step = step.difference(reduced_premises) # ignore axioms that we already found
        reduced_premises.update(step) # add new axioms

    return Problem(premises=reduced_premises, conjecture=problem.conjecture)
```



Parameter

Dienstag, 21. Januar 2020 14:18

Der Algorithmus kann auf viele Arten variiert werden:

- TF-IDF: term frequency, binary frequency ...
inverse document frequency, probabilistic idf ...
- LSI: Anzahl Dimensionen
(oder LSI ganz weglassen)
- Anzahl Iterationen (Spezialfall 1)
- Abschlusskriterium für Querying

```
class PremiseNumberAtLeast:  
    """Triggers after a fixed number of premises."""  
  
class ScoreBelow:  
    """Triggers when the premises fall below a given score."""  
  
class PremisePercentageAtLeast:  
    """Triggers when a given percentage of all premises has been selected."""  
  
class ScorePercentageAtLeast:  
    """Triggers when a given percentage of the cumulated similarity scores has been selected."""  
  
class All:  
    """Triggers when all of the given functions are triggered."""  
  
class Any:  
    """Triggers when any of the given functions is triggered."""  
  
Any(ScoreBelow(0.1), PremiseNumberAtLeast(10))
```

Evaluation

Dienstag, 21. Januar 2020 14:26

Von 4 Parametern je 2 Varianten:

- [0] term frequency vs. binary frequency [0]
- [1] kein LSI vs. LSI ($k=2\infty$) [1]
- [2] 1 Iteration vs. 2 Iterationen [2]
- [3] score $> 10^{-1}$ vs. score $> 10^{-8}$ [3]

0000	0001	0010	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

0000	0001	0010	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

0000	0001	0010	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

0 | 1
2 | 3

0000	0001	0010	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

Ergänzte: Selection ratio (proof steps)

	big	BIO002+1	NLP218+1	NLP261+1	NUM006+1	problem1	SET020+1	SET084+1	SET093+1	SET095+1
#axioms	10006	9161	71	1026860		18	3	43	43	47
identity	10006 31642	9161	71	1026860		18	3 15	43	43 56744	47 586
sine	3	2	47	72 212		18	3 15	3 35	0	4
irsel	14 51	20	40	11		15	3 15	26 2632	37 4434	32 287
irsel0000	2	4	24	67		15	1	10 148	6	10
irsel0001	2	100	28	100		15	1	37	40 4674	44 512
irsel0010	5	18	18	31		15	3 15	26 2632	25 262	29 220
irsel0011	13	72	18	43		15	3 15	26 2632	35 3391	32 287
irsel0100	2	100	24	100		15	1	10 148	6	10
irsel0101	4	100	33	100		15	2	40	40 4674	45 570
irsel0110	5	21	18	10		15	3 15	26 2632	25 262	29 220
irsel0111	14 51	21	40	10		15	3 15	26 2632	37 4434	32 287
irsel1000	2	4	27	67		15	1	10 148	6	10
irsel1001	2	100	28	100		15	1	37	40 4674	44 512
irsel1010	5	18	20	31		15	3 15	26 2632	25 262	29 220
irsel1011	13	72	20	43		15	3 15	26 2632	35 3391	32 287
irsel1100	2	100	27	100		15	1	10 148	6	10
irsel1101	2	100	29	100		15	2	38	40 4674	44 512
irsel1110	5	25	20	10		15	3 15	26 2632	25 262	29 220
irsel1111	13	25	36	10		15	3 15	26 2632	37 4434	32 287

→ TF-IDF, LSI (k=200),
2 Iterationen, Score > 10^-8

Ergebnisse (Parameter)

selector	id	sine	big	31642	BIO002	NLP218
0000 0001	0010	0011		51		
0100 0101	0110	0111				
1000 1001	1010	1011				
1100 1101	1110	1111				
NLP261	212		NUM006		problem1	15 15
					15 15	35
					15 15	148 2632
					15 15	2632 2632
					15 15	148 2632
					15 15	2632 2632
SET084	56744		SET093	586	SET020	35
4674 262	3391		512 220	287	148	2632 2632
4674 262	4434		570 220	287	148	2632 2632
4674 262	3391		512 220	287	148	2632 2632
4674 262	4432		512 220	287	148	2632 2632
SET095					2 Iterationen	
						Score > 0.1

2 Iterationen > 1 Iteration

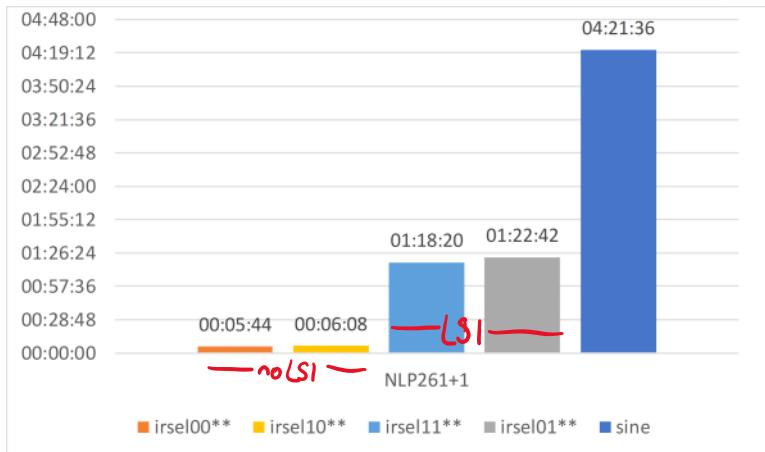
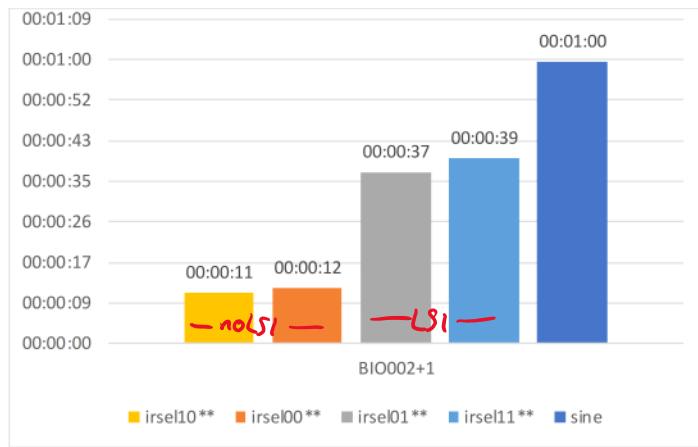
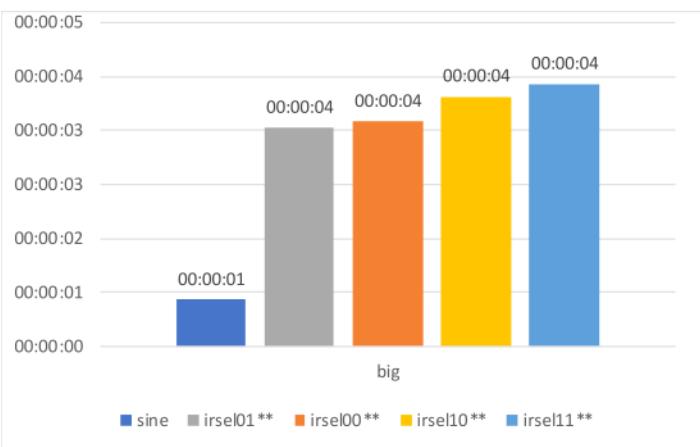
LSI ≈ kein LSI

\leftarrow Iterationen > 1 Iteration

LSI \approx kein LSI

TF-IDF \approx binary IDF

Ergebnisse (Zeit)



Python loops
vs.
Numpy arrays

SINE vs. irsel

NLP261+1

Ähnlichkeitsmaß für selektierte Axiome:

$$\text{Jaccard-Index}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

NLP261+1

selector1	selector2	Jaccard	both	only1	only2
sine	irsel0010	25.61%	21	51	10
sine	irsel1010	25.61%	21	51	10
sine	irsel0011	22.34%	21	51	22
sine	irsel1011	22.34%	21	51	22
sine	irsel0000	15.83%	19	53	48
sine	irsel1000	15.83%	19	53	48
sine	irsel0001	13.16%	20	52	80
sine	irsel1001	13.16%	20	52	80
sine	irsel	0.0%	0	72	11
sine	irsel0100	0.0%	0	72	100
sine	irsel0101	0.0%	0	72	100
sine	irsel0110	0.0%	0	72	10
sine	irsel0111	0.0%	0	72	10
sine	irsel1100	0.0%	0	72	100
sine	irsel1101	0.0%	0	72	100
sine	irsel1110	0.0%	0	72	10
sine	irsel1111	0.0%	0	72	10

$$\text{Jaccard}-\text{Index}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

SINE und irsel selection sehr unterschiedliche Axiome.

Score > 0.1

2 Iterationen

rein LSI

term frequency

Selektiert jeweils am ersten.

axiom1: $\forall [X, Y, Z]: ((\text{hypernym}(X, Y)) \wedge (\text{hypernym}(Y, Z))) \Rightarrow (\text{hypernym}(X, Z))$ - axiom2: $\forall [X, Y]: (\text{hypernym}(X, Y)) \Rightarrow (\text{hyponym}(Y, X))$ axiom3: $\forall [X, Y]: (\text{hyponym}(X, Y)) \Rightarrow (\text{hypernym}(Y, X))$ + kb10799: hypernym(n10488865, n10560637) + kb1046: lexicalization(n9986904, cytogeneticist) kb1047: lexical_category(n9986904, lexcat_noun) + kb1048: has_gloss(n9986904, 'a geneticist who specializes in the cellular component') kb1049: hypernym(n9986904, n10126424) - kb117072: derivation(n10560637, n5999797) + kb117076: hyponym(n10560637, n10488865) kb117077: lexical_category(n10560637, lexcat_noun) kb117080: hyponym(n10560637, n10428004) + kb117084: hyponym(n10560637, n10264437) kb117087: hyponym(n10560637, n9855630) kb117088: hyponym(n10560637, n9913824) kb117091: hypernym(n10560637, n7846) kb117096: hyponym(n10560637, n10301261) - kb188190: lexical_category(n6037666, lexcat_noun) - kb188254: derivation(n6037666, n9855630) - kb188287: category_of(n6037666, n9855630) - kb188298: hyponym(n6037666, n6075527) - kb207250: derivation(n6000644, n10301261) - kb207287: lexical_category(n6000644, lexcat_noun) - kb207312: hypernym(n6000644, n5999797) - kb207354: has_category(n6000644, n5999797) - kb207361: category_of(n6000644, n10301261) - kb235078: derivation(n5999797, n10560637) - kb235084: hyponym(n5999797, n6000644) - kb235086: meronym_part(n5999797, n5993844) - kb235089: lexical_category(n5999797, lexcat_noun) - kb235091: hyponym(n5999797, n6136258) - kb235095: category_of(n5999797, n6000644) - kb235099: hypernym(n5999797, n5996646) - kb235101: hyponym(n5999797, n6172789) - kb289397: hypernym(n6075527, n6037666) - kb289406: lexical_category(n6075527, lexcat_noun) + kbl53319: lexical_category(n8441203, lexcat_noun) kb289430: derivation(n6075527, n10126424) - kb319180: derivation(n9913824, n6084469) - kb319186: lexical_category(n9913824, lexcat_noun) - kb319202: has_category(n9913824, n6084469) kb319246: hypernym(n9913824, n10560637) + kb326589: means(cytogeneticist, n9986904)	only sine only irsel	kb340308: lexical_category(n10126424, lexcat_noun) kb340311: hyponym(n10126424, n9986904) - kb359780: hyponym(n5996646, n6123363) - kb359781: lexical_category(n5996646, lexcat_noun) - kb359787: hyponym(n5996646, n5999797) kb412519: lexical_category(n10428004, lexcat_noun) kb412525: hypernym(n10428004, n10560637) - kb412527: has_category(n10428004, n6090869) - kb412580: derivation(n10428004, n6090869) - kb424819: lexical_category(n6136258, lexcat_noun) - kb424836: hypernym(n6136258, n5999797) - kb479608: holonym_part(n5993844, n5999797) - kb479630: lexical_category(n5993844, lexcat_noun) - kb485451: hypernym(n6172789, n5999797) - kb485474: lexical_category(n6172789, lexcat_noun) - kb61946: derivation(n6090869, n10428004) - kb61968: category_of(n6090869, n10428004) - kb62028: lexical_category(n6090869, lexcat_noun) - kb78168: derivation(n6084469, n9913824) - kb78175: lexical_category(n6084469, lexcat_noun) - kb78225: category_of(n6084469, n9913824) + kb596055: lexical_category(n8860123, lexcat_noun) kb807143: hypernym(n10301261, n10560637) - kb807145: lexical_category(n10301261, lexcat_noun) - kb807160: derivation(n10301261, n6000644) - kb807162: has_category(n10301261, n6000644) + kb827864: hypernym(n10264437, n10560637) kb87554: lexical_category(n7846, lexcat_noun) kb87666: hyponym(n7846, n10560637) - kb886098: lexical_category(n9855630, lexcat_noun) - kb886101: has_category(n9855630, n6037666) - kb886103: derivation(n9855630, n6037666) kb886122: hypernym(n9855630, n10560637) kb886125: hyponym(n9855630, n10126424) - kb994304: hypernym(n6123363, n5996646) - kb994347: lexical_category(n6123363, lexcat_noun) + kb913165: lexical_category(n8524735, lexcat_noun)
---	-------------------------	--

```

- kb319186: lexical_category(n9913824, lexcat_noun)
- kb319202: has_category(n9913824, n6084469)
  kb319246: hypernym(n9913824, n10560637)
+ kb326589: means(cytogeneticist, n9986904)
  kb340303: derivation(n10126424, n6075527)
  kb340304: hypernym(n10126424, n9855630)

kb886125: hyponym(n9855630, n10126424)
- kb994304: hypernym(n6123363, n5996646)
- kb994347: lexical_category(n6123363, lexcat_noun)
+ kb913165: lexical_category(n8524735, lexcat_noun)

```

% Problem: Cytogeneticist is a hyponym of scientist
`fof(hypernym_transitiviy_1, conjecture, (hypernym(n9986904, n10560637))).`

NUM 006 + 1

selector1	selector2	Jaccard	both	only1	only2
sine	irsel	83.33%	15	3	0
sine	irsel0000	83.33%	15	3	0
sine	irsel0001	83.33%	15	3	0
sine	irsel0010	83.33%	15	3	0
sine	irsel0011	83.33%	15	3	0
sine	irsel0100	83.33%	15	3	0
sine	irsel0101	83.33%	15	3	0
sine	irsel0110	83.33%	15	3	0
sine	irsel0111	83.33%	15	3	0
sine	irsel1000	83.33%	15	3	0
sine	irsel1001	83.33%	15	3	0
sine	irsel1010	83.33%	15	3	0
sine	irsel1011	83.33%	15	3	0
sine	irsel1100	83.33%	15	3	0
sine	irsel1101	83.33%	15	3	0
sine	irsel1110	83.33%	15	3	0
sine	irsel1111	83.33%	15	3	0

Bei der Goldbach conjecture
 hat die Wahl der Parameter
 gar keinen Einfluss.

eines der 3 Axiome,
 das nie ausgewählt wird:
`fof(axiom_6, axiom,(! [X21,X22] : (~ id(X21,X22) | id(X22,X21)))).`

→ untl., da "id" eine geige idf hat.