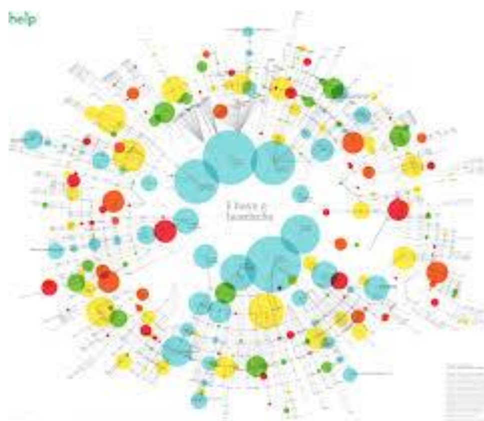
Pinpointing Ambiguity and Incompleteness in Requirements Engineering via Information Visualization and NLP



Fabiano Dalpiaz, Ivor van der Schalk, Garm Lucassen

Requirements Engineering Lab
Utrecht University, the Netherlands
March 22, 2018

1. Context and Motivation

1. Context and Motivation

- Requirements defects are still present in practice
  - Ambiguity, vagueness, incompleteness, etc.

The system shall send a message to the receiver, and it provides an acknowledge message within some seconds

[Rosadini 2017] [Vogelsang 2016]

1. Context and Motivation

- Requirements defects are still present in practice
  - Ambiguity, vagueness, incompleteness, etc.

The system shall send a message to the receiver, and <u>it</u> provides an acknowledge message within <u>some</u> seconds

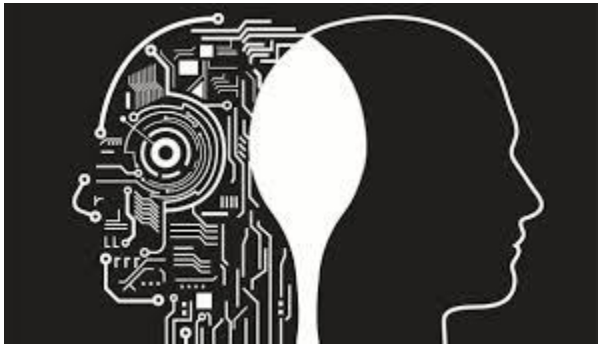
Referential pronoun ambiguity

Vague term

[Rosadini 2017] [Vogelsang 2016]

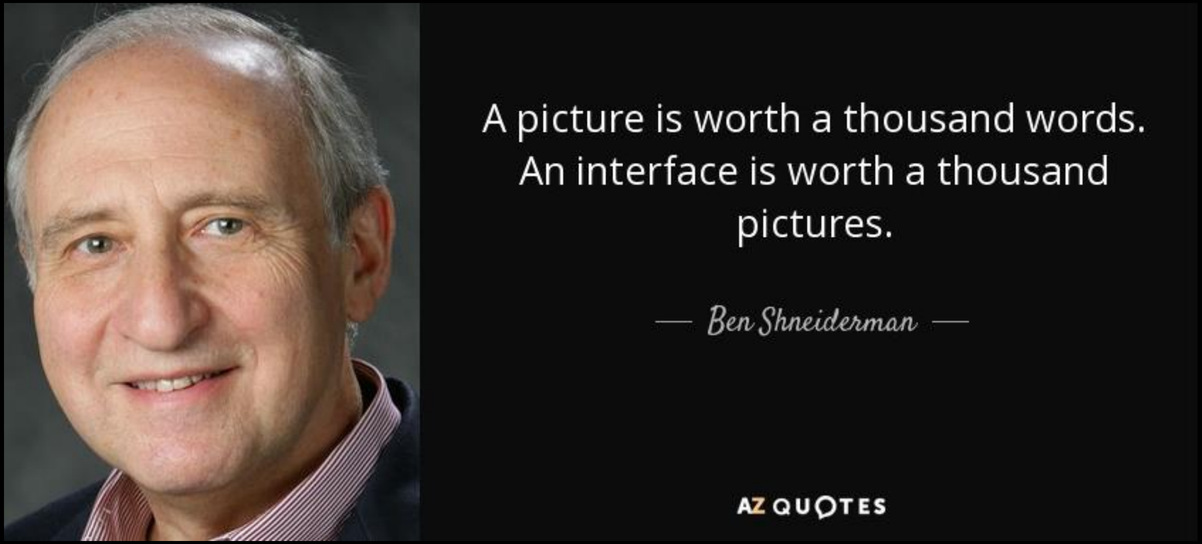
1. Context and Motivation

- Identifying requirements defects is still hard!
  - Natural language processing (NLP) tools do not deliver perfect accuracy in automated defect identification
  - Human analysts are effective, but how do they scale?



[Rosadini 2017] [Tjong 2013] [Vogelsang 2016]

2. Conceptual Solution



2. Conceptual solution

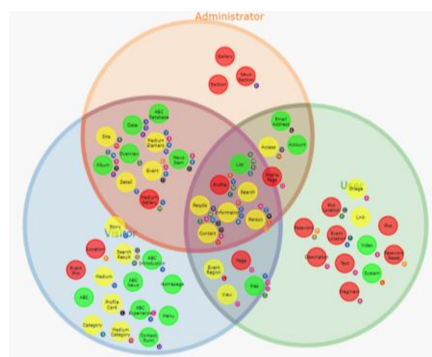
- Requirements artifact: user stories

  As a student,
  I want to receive my grades via e-mail,
  so that I can quickly check them.

Highly popular in agile dev! [Lucassen 2016]

- Idea: combine NLP with information visualization (InfoVis)
  → automation to help humans!

NLP



InfoVis

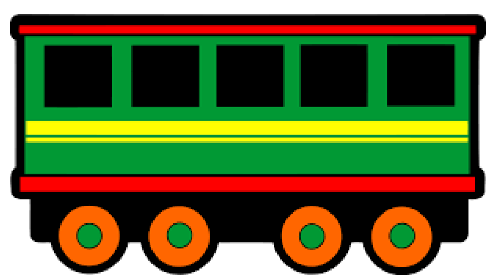
Human analyst

2. Conceptual solution

- Different stakeholders have their own viewpoints

- We focus on differences in their terminology!
  - For example, do *car* and *automobile* have the same meaning?
  - $[\![t]\!]^{V_1}$ is the denotation of term t according to viewpoint V_1

$[\![car]\!]^{V_{Fabiano}}$ $[\![car]\!]^{V_{train\,engineer}}$

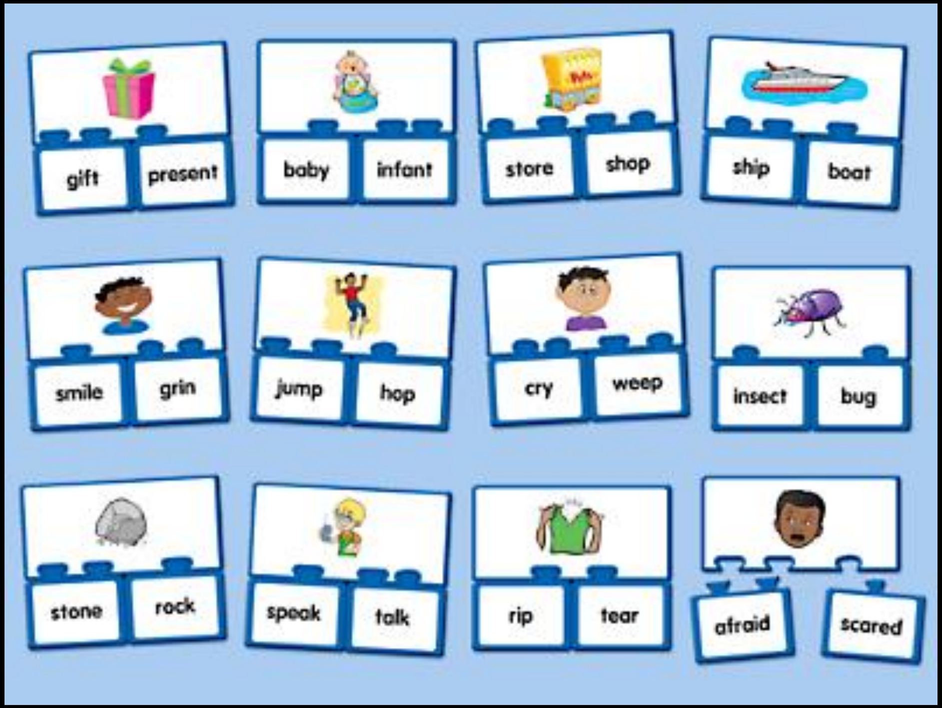


2. Conceptual solution

- We identify possible defects depending on the denotations that the viewpoints associate with a term

Relation [12]	Possible defect	Defect formalization	Example
Consensus	-	$[\![t_1]\!]^{V_1} = [\![t_1]\!]^{V_2}$	$[\![\text{bank}]\!]^{V_1} = \text{financial institution}$ $[\![\text{bank}]\!]^{V_2} = \text{financial institution}$
Correspondence	(Near-)synonymy leading to ambiguity	$[\![t_1]\!]^{V_1} = [\![t_2]\!]^{V_2}$	$[\![\text{car}]\!]^{V_1} = \text{road vehicle}$ $[\![\text{automobile}]\!]^{V_2} = \text{road vehicle}$
Conflict	Homonymy leading to ambiguity	$[\![t_1]\!]^{V_1} \neq [\![t_1]\!]^{V_2}$	$[\![\text{bank}]\!]^{V_1} = \text{financial institution}$ $[\![\text{bank}]\!]^{V_2} = \text{land alongside river}$
Contrast	Incompleteness	$[\![t_1]\!]^{V_1} \neq \bot \wedge [\![t_1]\!]^{V_2} = \bot$	$[\![\text{bank}]\!]^{V_1} = \text{financial institution}$ $[\![\text{bank}]\!]^{V_2} = \bot$

3. (Near-)Synonymy Detection

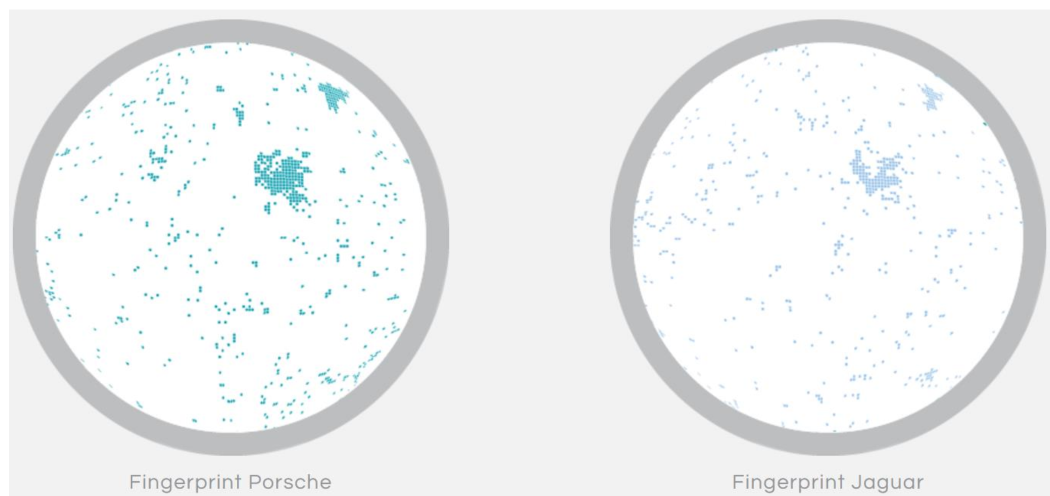


3. (Near-)Synonymy Detection

- Goal: identifying possible inter-view ambiguity

(Near-)synonymy leading to ambiguity	$[\![t_1]\!]^{V_1} = [\![t_2]\!]^{V_2}$	$[\![\text{car}]\!]^{V_1} = \text{road vehicle}$ $[\![\text{automobile}]\!]^{V_2} = \text{road vehicle}$

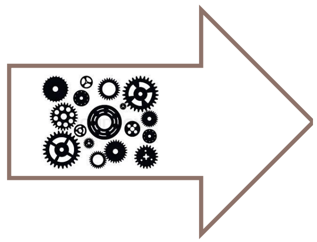
- How? We use Semantic Folding Theory (SFT)
  - Every term is associated a semantic fingerprint
  - Such fingerprints are created by analyzing huge amounts of text
  - Similar fingerprints indicate similar terms



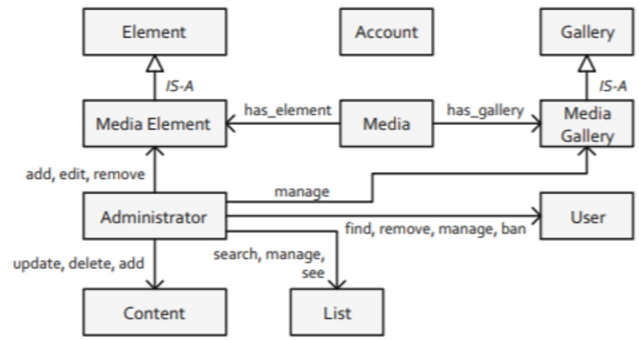
3. (Near-)Synonymy Detection

- How do we apply SFT to requirements engineering?

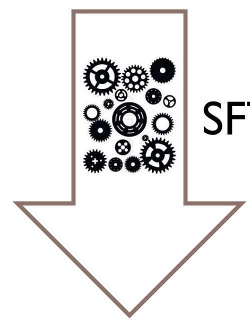
User Story Set
US1
US2
…
USn



Visual Narrator (Robeer 2015)



Conceptual model of the terms



SFT

Near-synonyms, a source of ambiguity

3. (Near-)Synonymy Detection

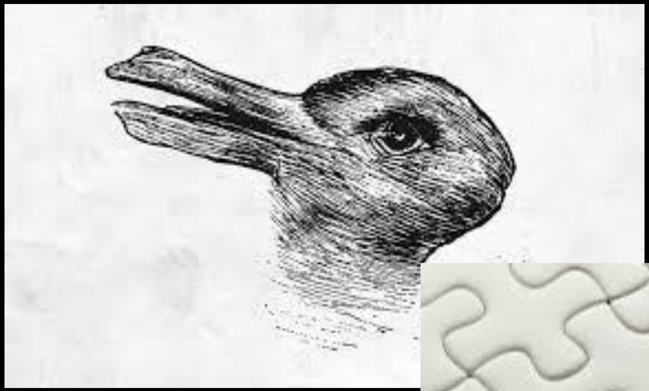
- **(Near-)synonymity between two terms t_1 and t_2**

$$ambig_{t1,t2} = \frac{2 \cdot sim_{t1,t2} + simc_{t1,t2}}{3}$$

  - A combination of term similarity and context similarity
  - 2/3 term similarity (car-automobile, etc.)
  - 1/3 context similarity: user stories where the terms appear
    - As a user, I want to make a bid for a car, so that …
    - As a visitor, I want to see the automobiles on the market, so that…

  - Weights assessed via a correlation study with humans

4. InfoVis for Ambiguity and Incompleteness



4. InfoVis for Ambiguity and Incompleteness

- NLP cannot (yet?) replace humans!
- Use InfoVis using Schneiderman's mantra

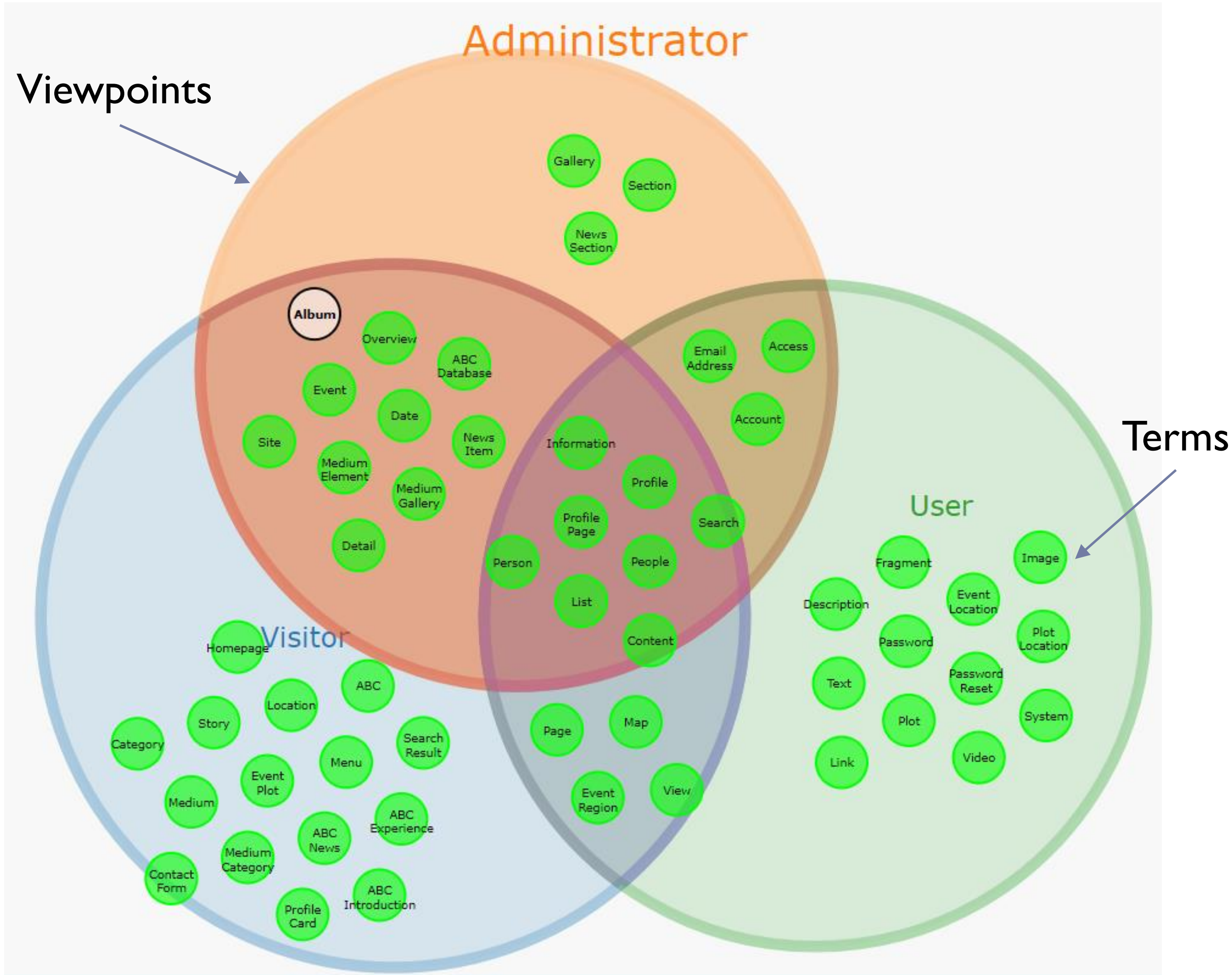
**Overview first, zoom and filter,
then details-on-demand**

- Focus mostly on ambiguity and incompleteness

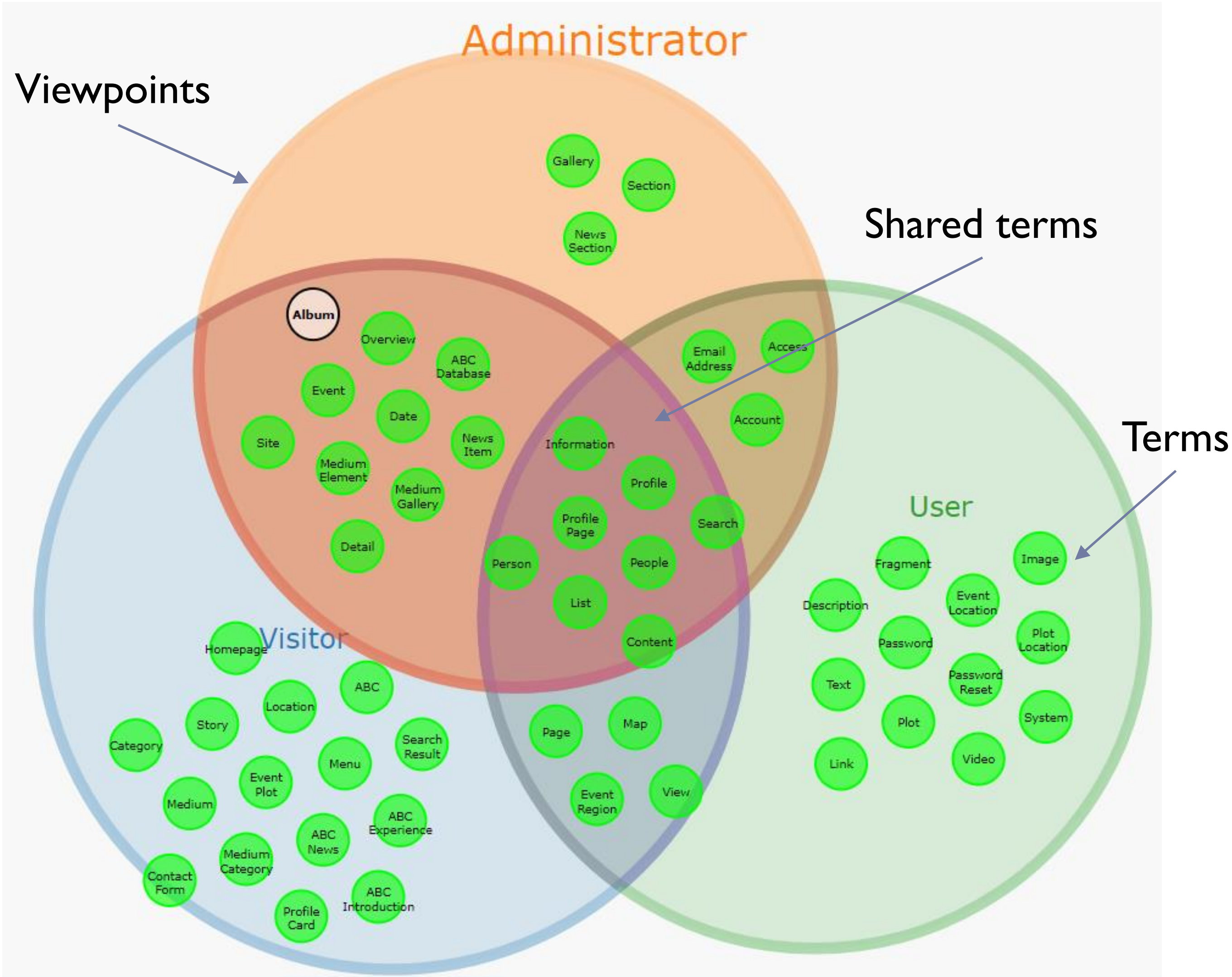
(Near-)synonymy leading to ambiguity	$[\![t_1]\!]^{V_1} = [\![t_2]\!]^{V_2}$	$[\![\text{car}]\!]^{V_1} = \text{road vehicle}$ $[\![\text{automobile}]\!]^{V_2} = \text{road vehicle}$

Incompleteness	$[\![t_1]\!]^{V_1} \neq \perp \wedge [\![t_1]\!]^{V_2} = \perp$	$[\![\text{bank}]\!]^{V_1} = \text{financial institution}$ $[\![\text{bank}]\!]^{V_2} = \perp$

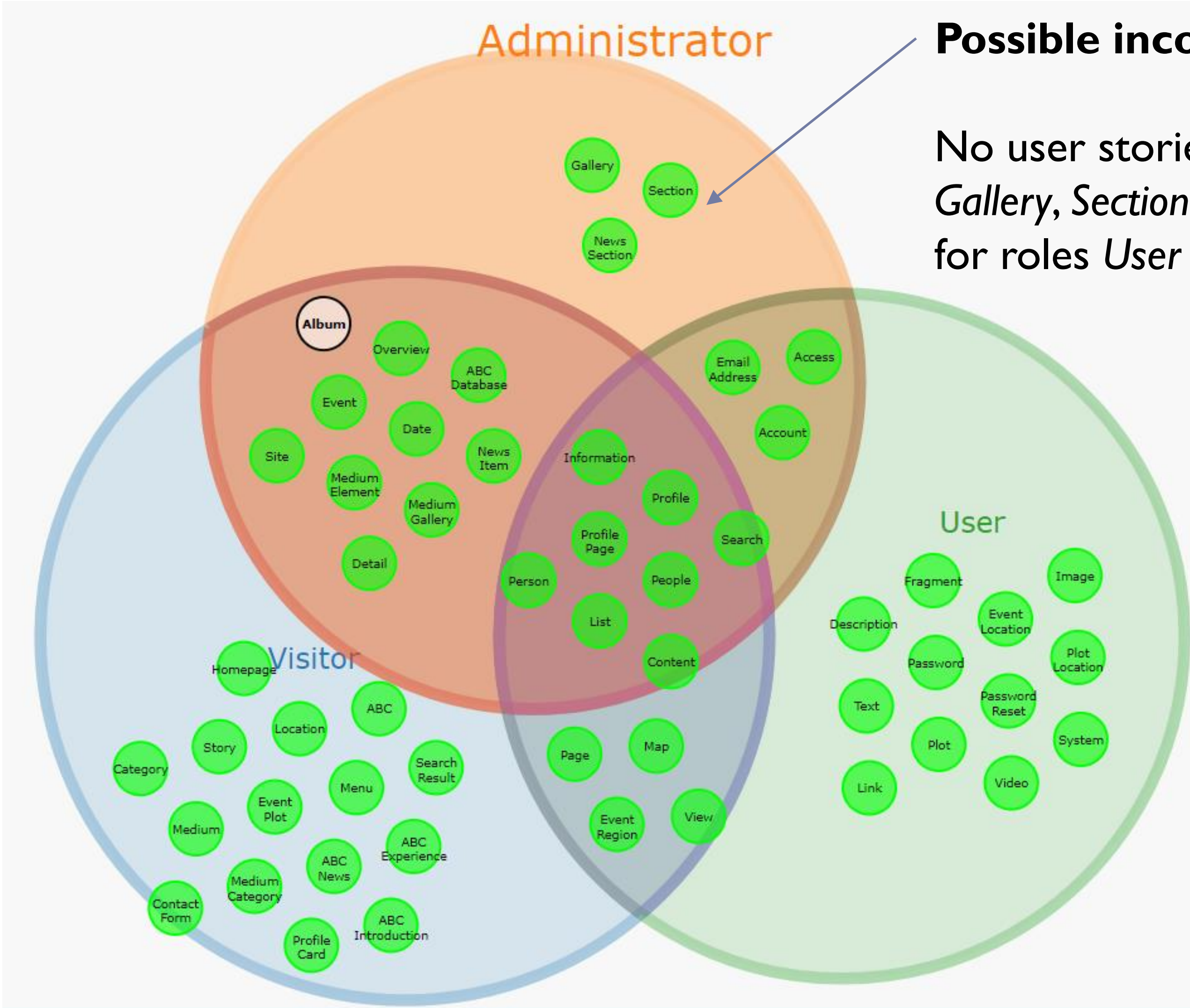
4. InfoVis for Ambiguity and Incompleteness



4. InfoVis for Ambiguity and Incompleteness



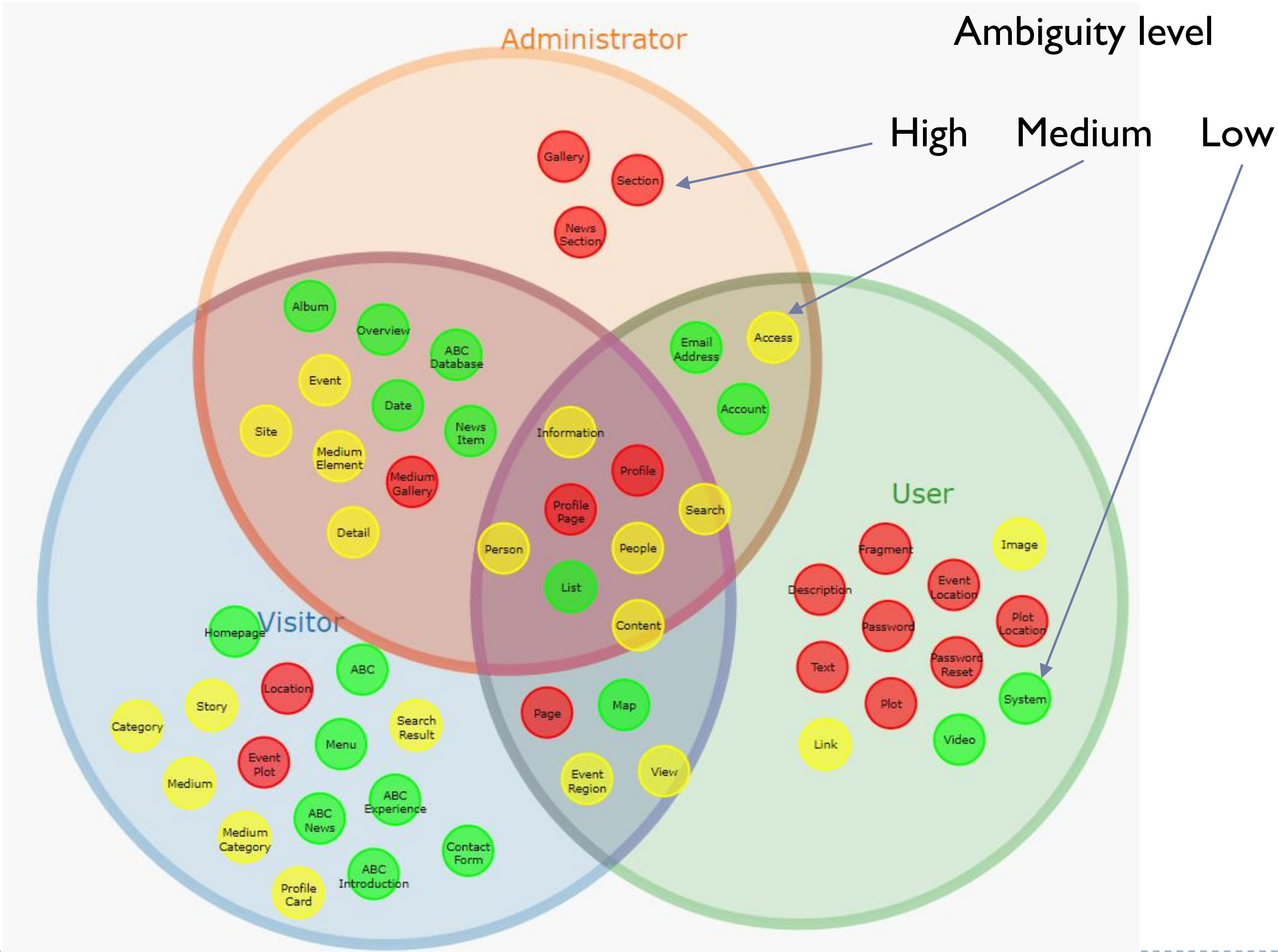
4. InfoVis for Ambiguity and Incompleteness



Possible incompleteness

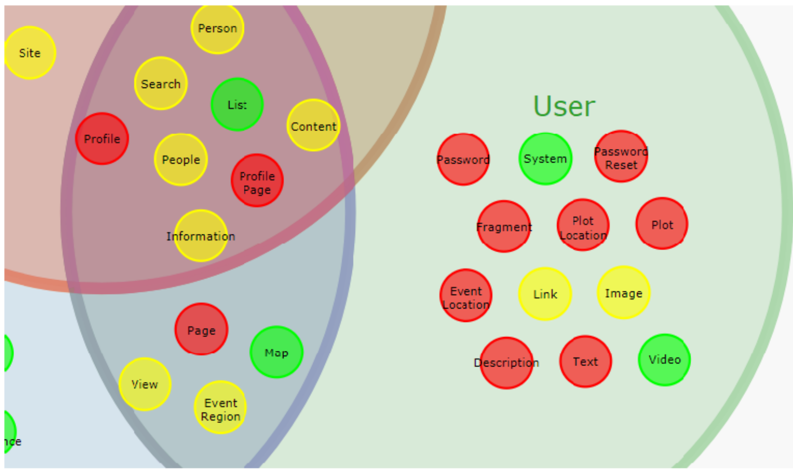
No user stories about *Gallery*, *Section*, *News Section* for roles *User* and *Visitor*?

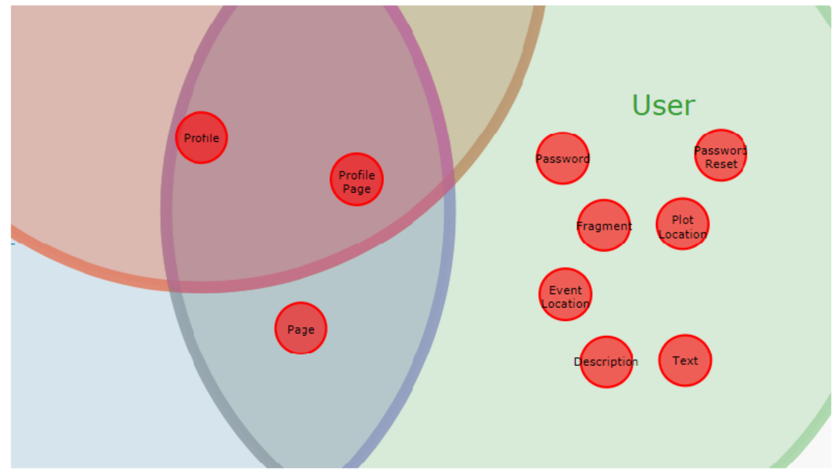
4. InfoVis for Ambiguity and Incompleteness



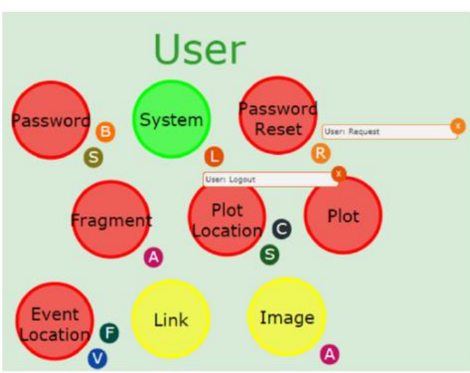
4. InfoVis for Ambiguity and Incompleteness

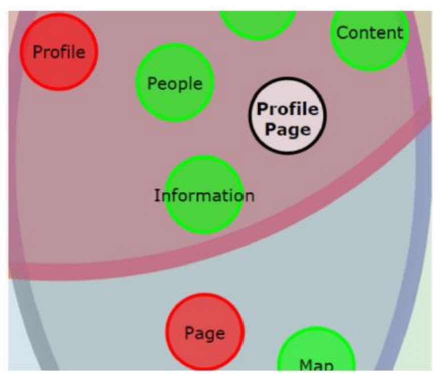
- Filter

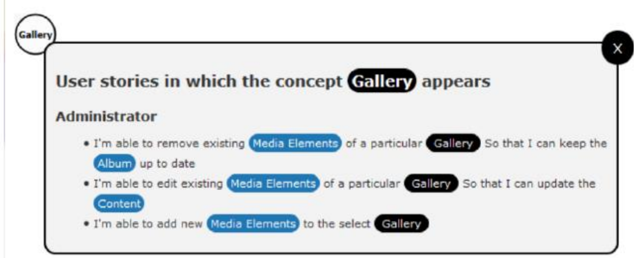




- Zooming







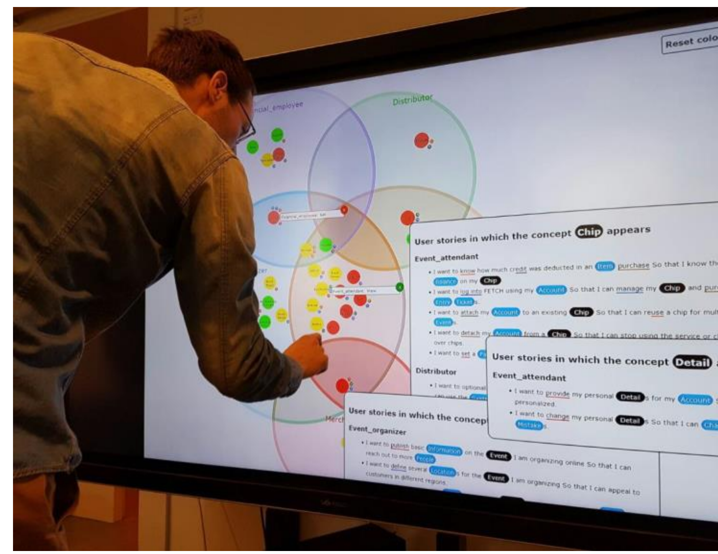
5. Quasi-Experiment

5. Quasi-Experiment

- Hypothesis: analysts who use our approach obtain a significantly higher…
  - precision in finding ambiguities (H1);
  - recall in finding ambiguities (H2);
  - precision in finding missing requirements (H3);
  - recall in finding missing requirements (H4);
- …compared to analysts using a pen-and-paper inspection.

5. Quasi-Experiment

- Study purpose/object: compare the relative effectiveness of
  - Our approach (REVV tool) supported by an 84” touch screen
  - A manual, pen-and-paper inspection of the requirements



- With voluntary MSc students in information science (n=8)
- 2 groups of 2 students with REVV
- 2 groups of 2 students pen&paper

5. Quasi-Experiment

- Constructs were defined through brainstorming among the authors, a pilot test, and the existing literature

- A **missing user story** is one whose absence inhibits the realization of at least another user story
- An **ambiguity** occurs when two user stories contain distinct terms that shares the same denotations

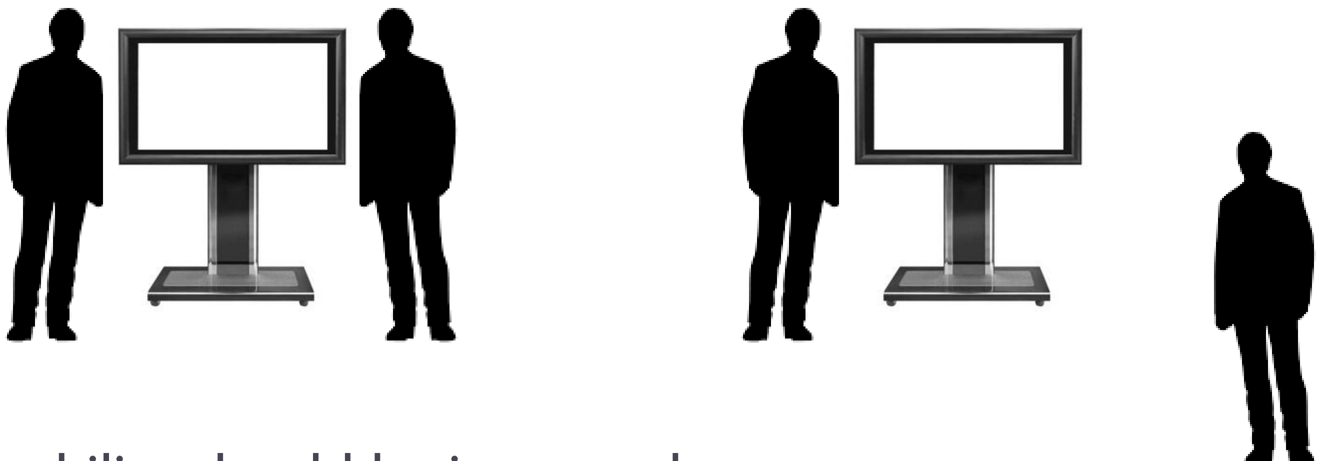
5. Quasi-Experiment

- Quantitative results
  - Reject H1 and H3 (precision)
  - Retain H2 and H4 (recall)

	Total TP	#TP	#FP	Precision	Recall
Session 1 – ambiguity					
Pen & paper	28	8	1	0.888	0.285
Tool		23	4	0.851	0.821
Session 2 – ambiguity					
Pen & paper	12	3	4	0.428	0.25
Tool		9	0	1	0.75
Session 1 – incompleteness					
Pen & paper	9	4	1	0.8	0.444
Tool		5	2	0.714	0.555
Session 2 – incompleteness					
Pen & paper	5	2	2	0.5	0.4
Tool		3	2	0.6	0.6

5. Quasi-Experiment

- Qualitative findings
  - Different types of interaction with the screen



  - Tool usability should be improved
  - The tool can lead to time savings

6. Discussion and Outlook

6. Discussion and outlook

- A first attempt to combine NLP and InfoVis
- Focus on ambiguity (near-synonymity) and missing reqs
- Inspiration by Venn diagrams

- Future directions
  - Algorithm can be further tuned (risk of overfitting?)
  - Evaluation, evaluation, evaluation!
  - Using domain ontologies for better results?

Thanks from the Requirements Engineering Lab at Utrecht University!

Fabiano Dalpiaz
Sjaak Brinkkemper
Marcela Ruiz
F. Basak Aydemir
Sietse Overbeek
Gerard Wagenaar
Davide Dell'Anna
Govert-Jan Slob