Relational Learning and Feature Extraction by **Querying** over **Heterogeneous Information Networks**

Parisa Kordjamshidi, *Tulane University; Florida Institute of Human and Machine Cognition*
pkordjam@tulane.edu
Sameer Singh, *University of California, Irvine*
Daniel Khashabi, *University of Illinois at Urbana-Champaign*
Christos Christodoulopoulos, *Amazon Research Cambridge, UK*
Mark Summons, *University of Illinois at Urbana-Champaign*
Saurabh Sinha, *University of Illinois at Urbana-Champaign*
Dan Roth, *University of Illinois at Urbana-Champaign*

Statistical Relational AI

August 2017, Sydney, Australia

Motivation

Motivation

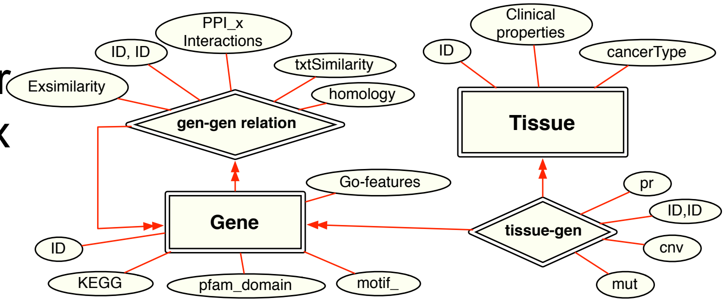
- **Real-world systems:** many of them use *heterogeneous information networks* that consist of numerous interacting components of different types.

Motivation

- **Real-world systems:** many of them use *heterogeneous information networks* that consist of numerous interacting components of different types.
- **Examples:** are Biological Networks, Social Networks or general Knowledge Graphs with arbitrary complex structures.

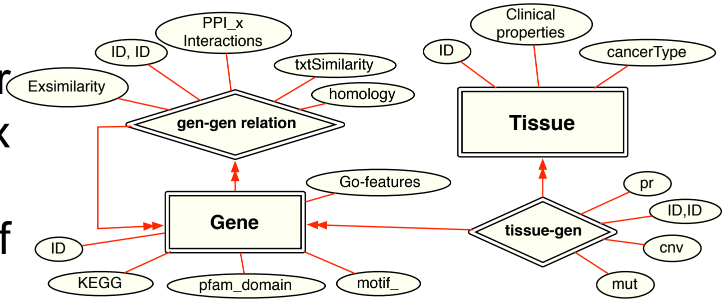
Motivation

- **Real-world systems:** many of them use *heterogeneous information networks* that consist of numerous interacting components of different types.
- **Examples:** are Biological Networks, Social Networks or general Knowledge Graphs with arbitrary complex structures.



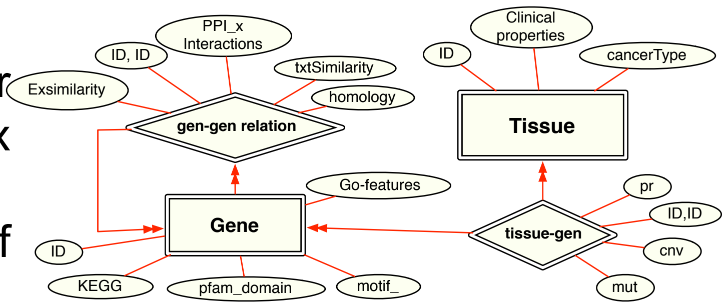
Motivation

- **Real-world systems:** many of them use *heterogeneous information networks* that consist of numerous interacting components of different types.
- **Examples:** are Biological Networks, Social Networks or general Knowledge Graphs with arbitrary complex structures.
- **Existing research:** addresses many challenges of working with such networks for various mining tasks. Mostly specialized approaches perform specific types of analysis over a network design with **task specific implementations**.



Motivation

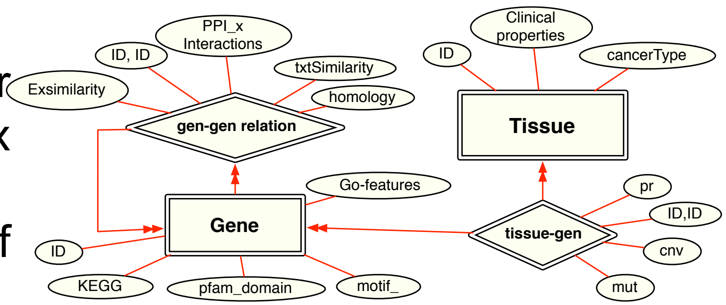
- **Real-world systems:** many of them use *heterogeneous information networks* that consist of numerous interacting components of different types.
- **Examples:** are Biological Networks, Social Networks or general Knowledge Graphs with arbitrary complex structures.
- **Existing research:** addresses many challenges of working with such networks for various mining tasks. Mostly specialized approaches perform specific types of analysis over a network design with **task specific implementations**.



- **Demand:** A general solution for **easily constructing** such networks or **systematically manipulating** them by **various analysis units based on learning and reasoning**

Motivation

- **Real-world systems:** many of them use *heterogeneous information networks* that consist of numerous interacting components of different types.
- **Examples:** are Biological Networks, Social Networks or general Knowledge Graphs with arbitrary complex structures.
- **Existing research:** addresses many challenges of working with such networks for various mining tasks. Mostly specialized approaches perform specific types of analysis over a network design with **task specific implementations**.



- **Demand:** A general solution for **easily constructing** such networks or **systematically manipulating** them by **various analysis units based on learning and reasoning**
- **Demand:** Relational data **representation**, **learning** and flexible intelligent data analysis, as well as the **evolution** of these networks based on the analysis outcomes, need to be placed in a well-defined framework.

Placing the Required Components in Saul

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.
- The query language, in addition to basic retrieval operations, helps in,

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.
- The query language, in addition to basic retrieval operations, helps in,
    - Preparing learning examples

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.
- The query language, in addition to basic retrieval operations, helps in,
    - Preparing learning examples
    - Feature extraction

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.
- The query language, in addition to basic retrieval operations, helps in,
    - Preparing learning examples
    - Feature extraction
    - Performing meta analysis on the graph

Placing the Required Components in Saul

Saul is a declarative Learning based programming language that targets **designing relational learning models.**

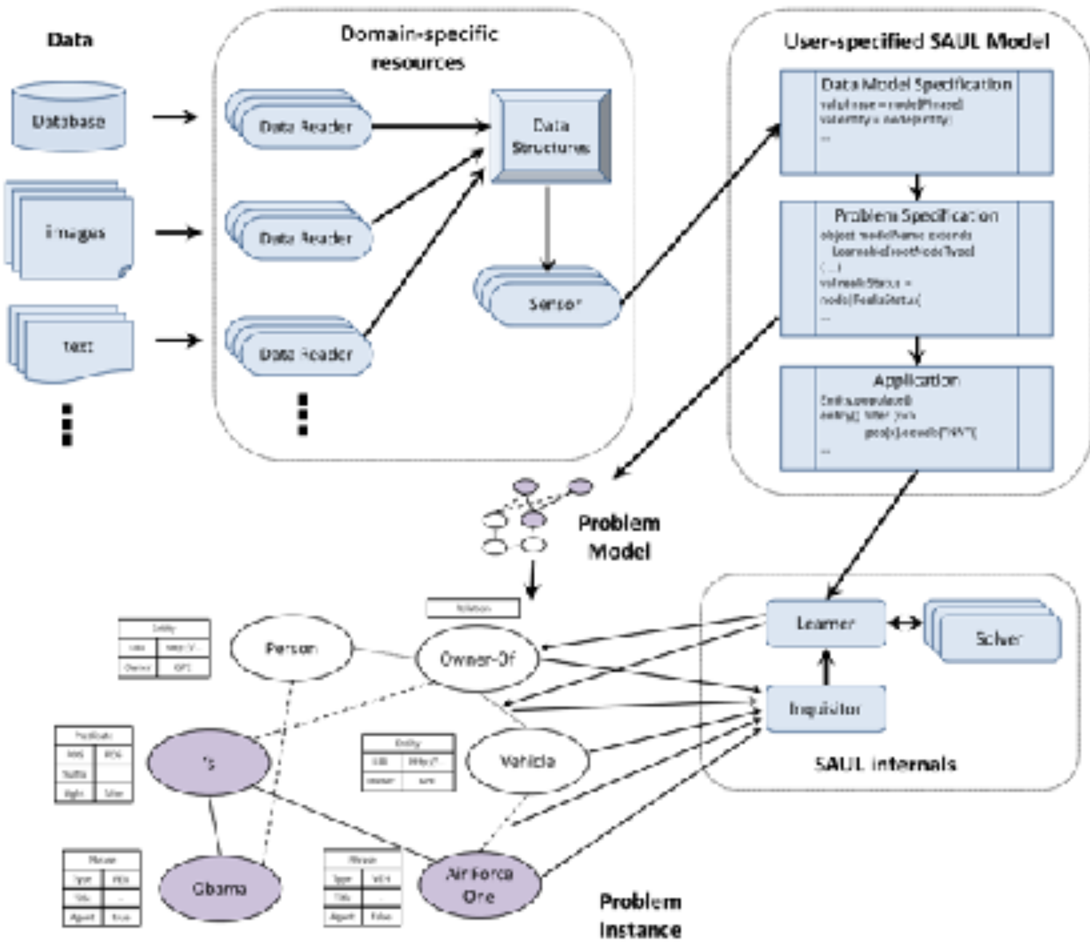
- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.
- The query language, in addition to basic retrieval operations, helps in,
    - Preparing learning examples
    - Feature extraction
    - Performing meta analysis on the graph
- Evolving the graph based on the outcomes of learning and inference

Placing the Required Components in Saul

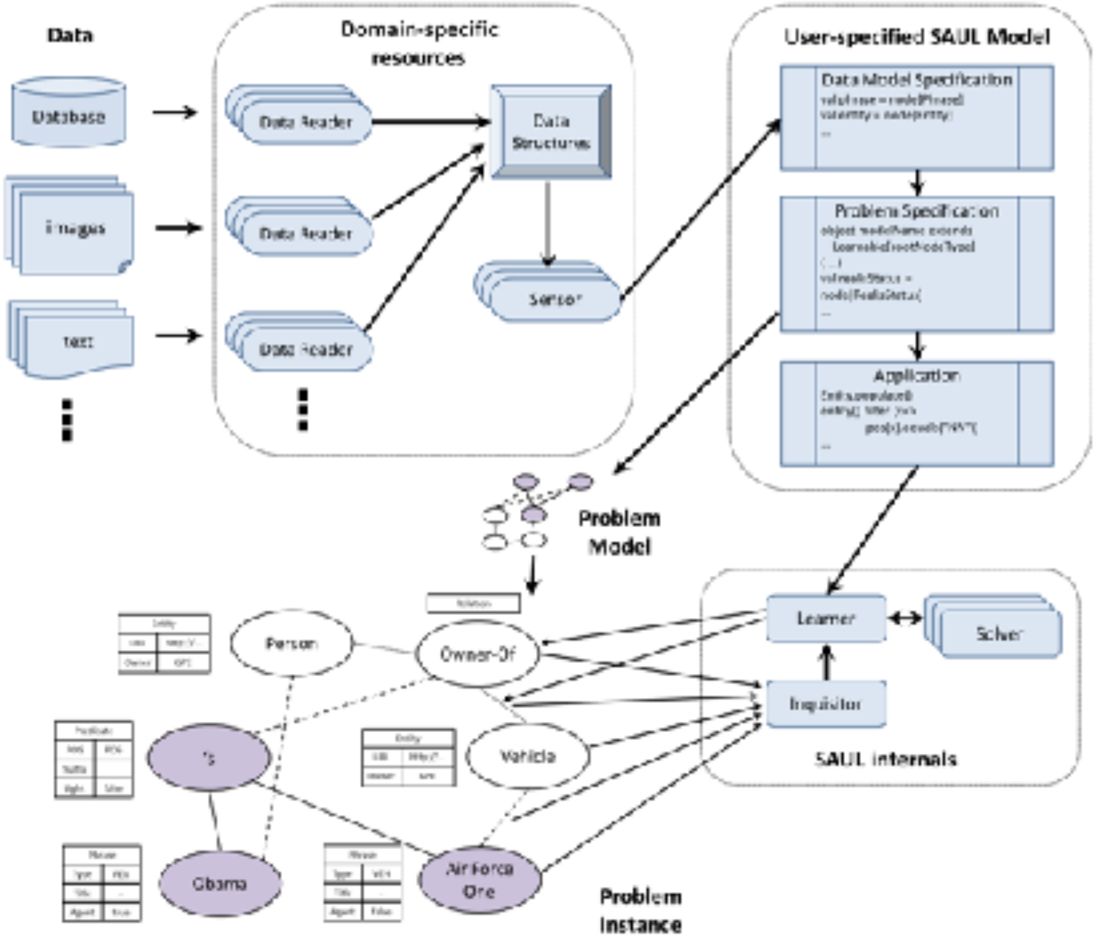
Saul is a declarative Learning based programming language that targets **designing relational learning models.**

- A language for specifying the schema of a graph —a relational *datamodel.*
- Interaction with messy, naturally occurring data using domain sensors.
- Integrating data from heterogeneous resources in one unified graph.
- Graph traversal queries operating on the *datamodel*.
- Seamless integration of learning and inference algorithms to manipulate the data.
- The query language, in addition to basic retrieval operations, helps in,
    - Preparing learning examples
    - Feature extraction
    - Performing meta analysis on the graph
- Evolving the graph based on the outcomes of learning and inference
- Considering expert knowledge at a very high level with a logical representation.

Overview of the components

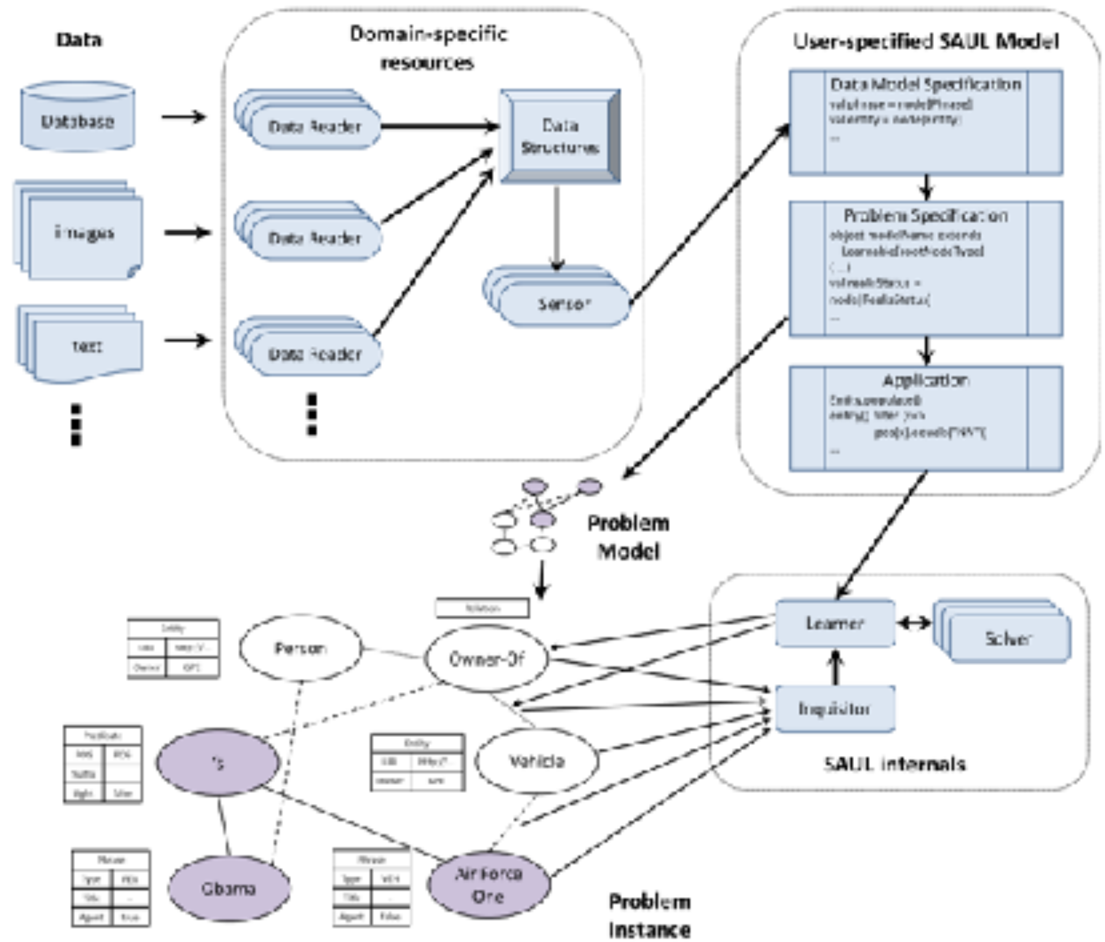


Overview of the components



- Examples?

Overview of the components



- Examples?

Come to the poster!