W4. Check List — Junyi

▼ How to integrate data

▼ how the data elements from a new data source should be added to the knowledge graph → schema mapping problem

▼ How does it work

- ssumes the existence of a schema which will be used for storing new data coming from another source → Schema mapping then defines which relations and attributes in the input database corresponds to which properties and relations in the knowledge graph
- techniques for bootstrapping schema mappings. The bootstrapped schema mappings can be post corrected through human intervention.

▼ challenges and arguing that one should be prepared for the eventuality that this process will be largely manual and labor-intensive

▼ (1) difficult to understand schema

Sometimes, the relation and attribute names do not have semantics (e.g., segment1, segment2) which do not lend themseves to any realistic automated prediction of the mappings.

▼ (2) complexity of mappings

Mappings may involve calculations, applying business logic, and taking into account special rules for handling situations such as missing values. It becomes a tall order to expect any automatic process to infer such complex mappings.

▼ (3) lack of training data available.

As the schema information, by definition, is much smaller than the data itself, it is unrealistic to expect that we will ever have large number of schema mappings available against which a mapping algorithm could be trained.

▼ an approach for specifying mappings between the schema of the input source and the schema of the knowledge graph

▼ linguistic matching

be used on the name of an attribute or on the text description of an attribute.

▼ matching based on instances

examines the kind of data that exists.

▼ matching based on constraints.

if the schema specifies that a particular attribute must be unique for an individual, and must be a number, it is a potential match for identification attributes such as an employee number or social security number.

▼ pratically difficult: a fully automated approach to schema mapping → bootrapping the schema mappings and validating them using human input

▼

▼ Recognizing if two instances refer to the same object in the real-world → record linkage problem

▼ tow-step approach: blocking and matching → significantly reducing the comparisons that must be performed.

▼ 1. **blocking step**

▼ involves a fast computation to select a subset of records from the source and the target that will be considered during a more expensive and precise matching step.

▼ Exampe: In the example considered above, we could use a blocking strategy that considers matching only those records that match on the state.

▼ 2.**matching step**

▼ we pairwise match the subset of records that were selected during blocking.

▼ 3.**Applying the Rules**

Once we have learned the rules, the next step is to apply them on the actual data.

- efficient application of rules through indexing

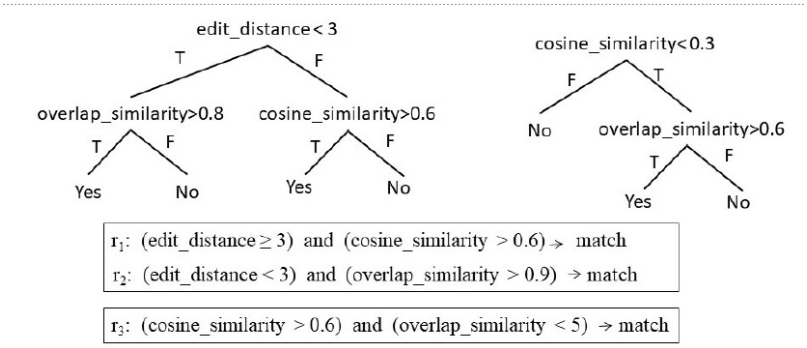
- e.g., If we have indexed the dataset on the size of the movies, it is efficient to choose only those movies whose sizes are between 2 and 4, and filter the set even further through the application of the blocking rule

▼ How to implement blocking and matching

▼ Both blocking and matching steps work by learning a random forest through an active learning
process.

▼ A random forest is a set of decision rules that gives its final prediction through a majority vote returned by individual rules.

- rely on similarity functions



▼ general principles for selecting similarity functions

- numeric-valued attributes → exact match, absolute difference, relative difference, and Levenstein distance
- String-valued attributes → edit distance, cosine similarity, Jaccard similarity, and TF/IDF functions

▼ Active learning is a learning process that constructs the random forest by actively monitoring its performance on the test data, and selectively choosing new training examples to iteratively improve its performance.

▼ Alogorithm

1. Randomly select pairs from the two data sets → ask the users to label them
2. use similarity functions to obtain features

3. learn random forest
4. apply the learned rules to new selected pairs → evaluate the rules
5. iterate
6. once the learning algorithm converges, present the rules to the user
7. Retain the rules validated by the user

▼ Distinctions between matching and blocking

- matching rules (as the final step) are more exact/price
- matching is usually verified through human intervention