BERTMap: A BERT-based Ontology Alignment System

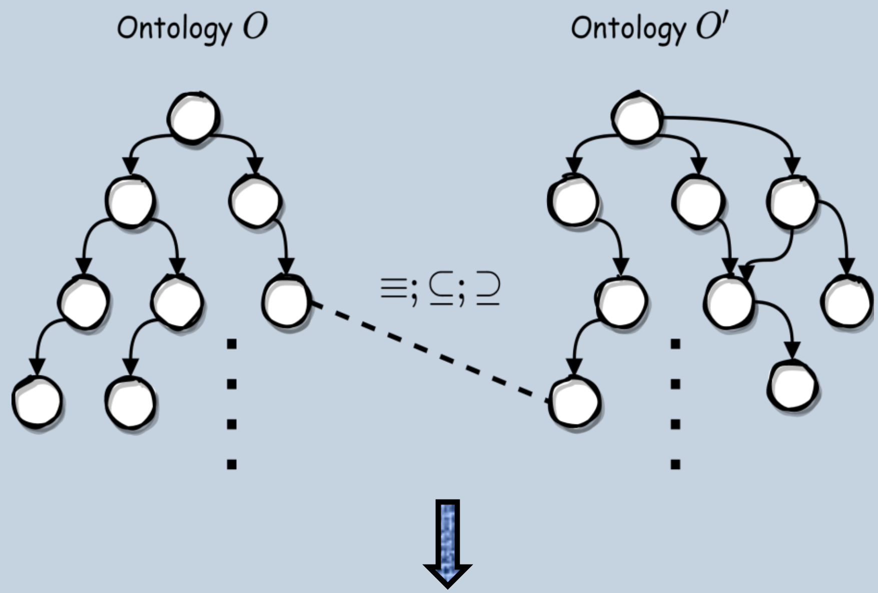
Yuan He[1], Jiaoyan Chen[1], Denvar Antonyrajah[2], Ian Horrocks[1]

Department of Computer Science, University of Oxford[1]

Samsung Research UK[2]

Ontology Alignment (or Matching)

- Entity:
  - Class
  - Property
  - Instance
- Relationship:
  - Equivalence
  - Subsumption



$$Mapping = \langle e \in O, e' \in O', rel, score \rangle$$

Challenges

- Disambiguation:
  - **Naming:**
    - E.g., the concept named *muscle layer* in SNOMED-CT is named *muscularis propria* in FMA.
  - **Contexts:**
    - E.g., in FoodOn, there are two concepts named *mushroom* that are categorized in both *Plant* and *Food.*
- Search space reduction:
  - Traversing all possible mappings takes $O(n^2)$.

Challenges

- Extreme **positive-negative imbalance**:
  - Matched entity pairs are often of *several orders fewer* than unmatched ones.
  - High-quality reference data is *often not available*, e.g., UMLS is considered as a silver standard (not necessarily *complete* or *correct*). This induces two further issues:
    - Results of automatic evaluation *lack reliability*.
    - Supervised learning setting is often *unfeasible*.

Related Work

- Classic OM systems:
  - **LogMap** uses its *lexical index* to create anchor mappings, then iteratively alternates between mapping discovery and repair.
  - **AML** relies on a *mixed matcher* that involves string similarity comparison and many other hand-crafted features.

- Machine learning-based OM solutions:
  - Most of them are *supervised* and rely on sufficiency of *annotated data*, and/or complicated *feature engineering*.
  - *Unsupervised* OM systems are still using traditional *non-contextual embeddings* (e.g., **Word2Vec**) to encode textual information.
  - **LogMap-ML** is the ML extension of **LogMap**, which uses the anchor mappings for training and **Word2Vec** to encode class labels.

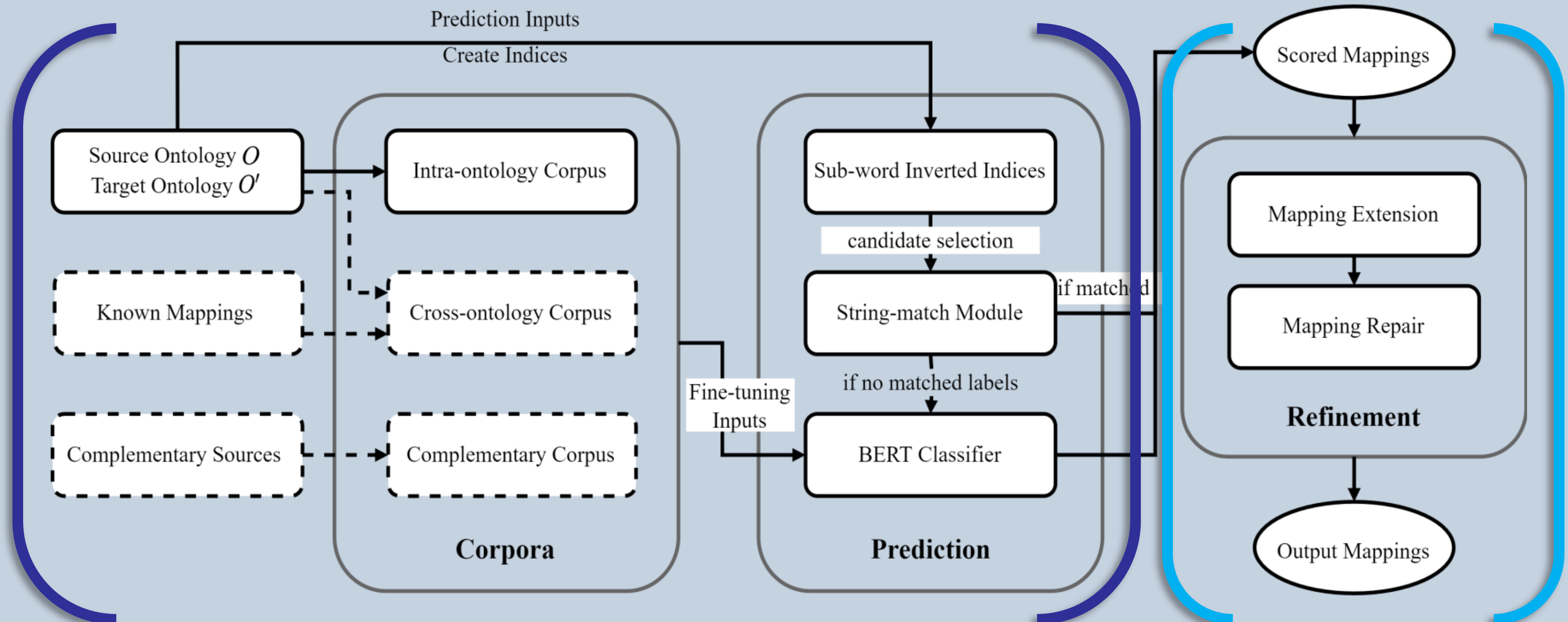
Related Work

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers:
  - **BERT** can compute dynamic (*contextual*) embeddings whereas the traditional word embedding systems (e.g., **Word2Vec**, **GloVe**) yield the static (*unified*) word vectors. For example,

    The bank robber was seen fishing on the river bank.

  - *Pre-train & Fine-tune* is a popular learning scheme for language models in NLP.
  - *Pre-trained* BERTs are widely accessible; often, only *fine-tuning* is needed and it takes *2-4 epochs* to achieve convergence.

Model Outline



Make predictions based on the ensemble results of class label (text) classification

Refine mappings through extension & repair

Mapping Prediction: Text Semantics Corpora

Ontologies → Synonyms & Non-synonyms → Text Semantics Corpora

Synonyms: An ontology class could have multiple *aliases* defined by some annotational properties, e.g., *rdfs:label*, *oboInOwl:hasExactSynonym*.

Note: We considered both *reversed* and *identity* synonyms

Non-synonyms are retrieved from *either* label pairs of two random classes (*soft*) *or* label pairs of logically disjoint classes (*hard*).

Mapping Prediction: Text Semantics Corpora

Intra-ontology (from within an ontology)

- As described in the previous slide.

Cross-ontology (from a small portion of given mappings)

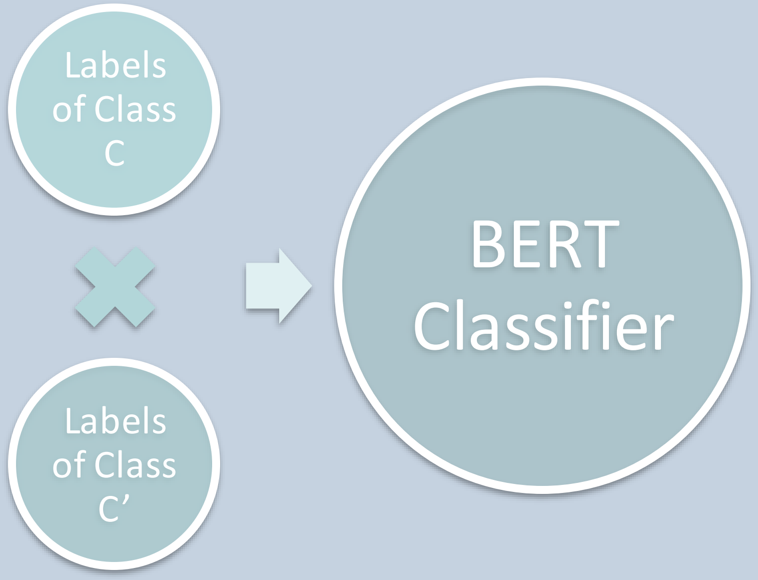
- *Synonyms* are extracted from label pairs of the matched classes.
- *Non-synonyms* are extracted from randomly aligned classes.

Complementary (from an auxiliary ontology)

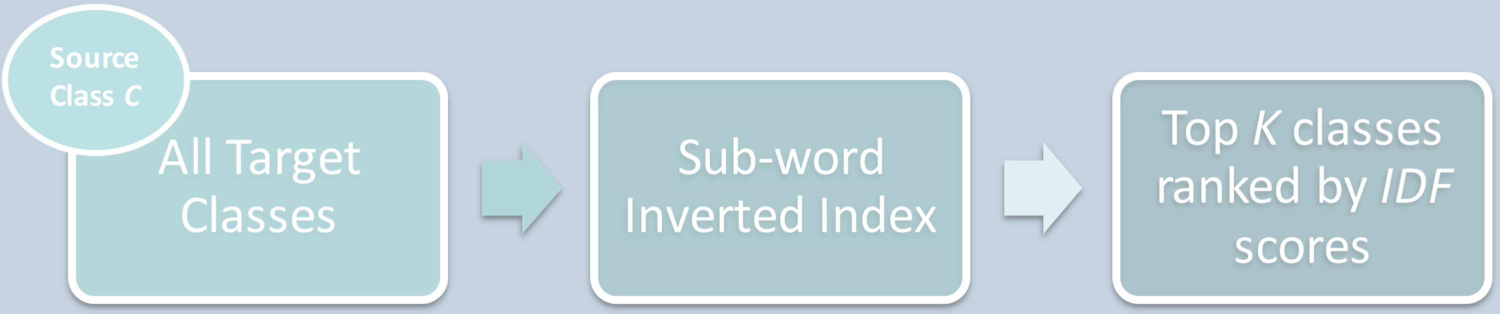
- Augmenting data for training the synonym classifier.

Mapping Prediction: Synonym Classifier

- Our *synonym classifier* comprises a **pre-trained BERT model** with an **extra binary classification layer**.
- We fine-tune the BERT classifier on the **text-semantics corpora** constructed from ontologies.
- The similarity score for class *C* and *C′* is computed using the **average of the synonym scores** of the paired labels of *C* and *C′*.



Mapping Prediction: Candidate Selection

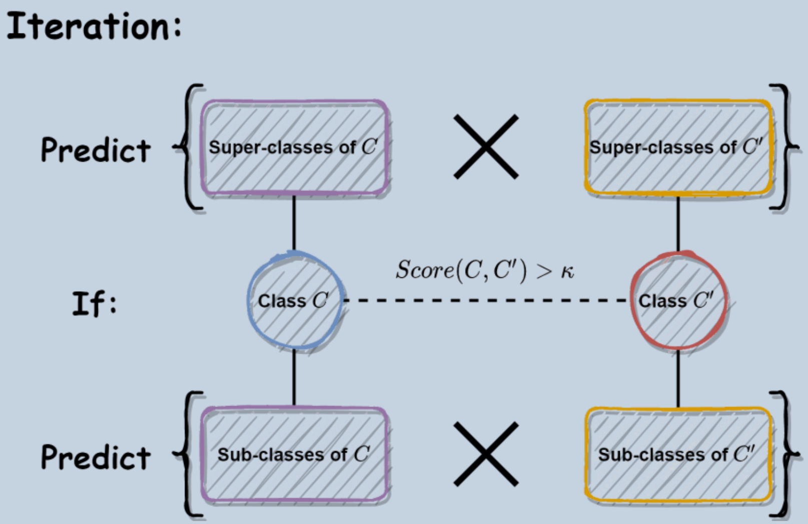


- For each source class *C*, we select top *K* ranked target classes according to **Sub-word Inverted Index**-based *IDF* scores.
- **Outcome**: Reducing the *quadratic* search space to *linear*.
- **Limitation**: Selected classes are assumed to have *at least one common sub-word token* with the given class.
- **Advantages:** various word forms can be captured; unknown words are treated as consecutive known sub-words.

Mapping Refinement: Extension

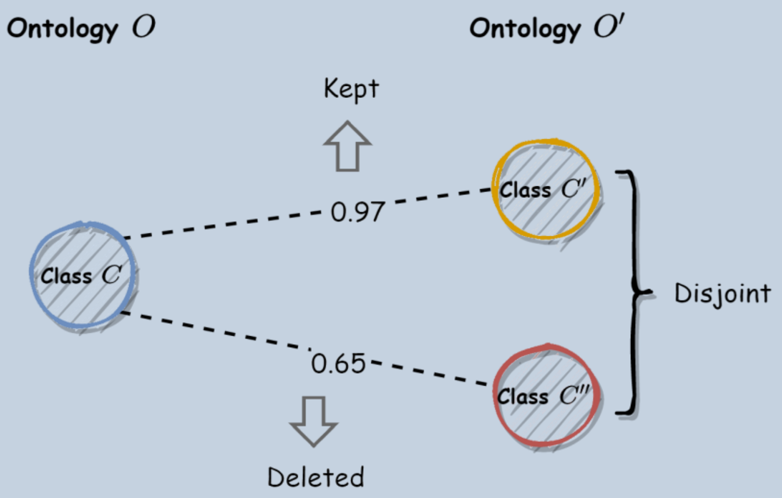
- **Locality Principle**: *Semantically related* classes of the matched classes are likely to be matched.

- **Outcome:** To *recall more mappings*, especially the ones that violate the assumption of token sharing (in candidate selection).



Mapping Refinement: Repair

- **Issue**: Reasoning over each input ontology may lead to *inconsistency* after alignment.
- **Solution**: Remove a minimal set of mappings (*a.k.a. diagnosis*) to achieve *consistency*.
- **Outcome:** To *improve precision* while keeping the recall as much as possible.



- We adopt the mapping repair tool built by Jimenez-Ruiz et al. (2013).

Experiment: Task Settings

FMA-NCI	FMA-SNOMED	FMA-SNOMED+
#Classes: 3696-6488	#Classes: 10157-13412	An extended task of FMA-SNOMED, where SNOMED is "updated" according to the most recent version.
#Refs (=): 2686	#Refs (=): 6026	
#Refs (?): 338	#Refs (?): 2982	

- *FMA-NCI* and *FMA-SNOMED* are extracted from the LargeBio track of OAEI.
- We evaluate all the models by Precision, Recall, and F1, subject to the negligence of mappings in *Refs (?)*.

Experiment: Results

Hyper-params | Unsupervised | Semi-supervised

Ablations / Baselines

System	$\{\tau, \lambda\}$	90% Test Mappings Precision	Recall	Macro-F1	70% Test Mappings Precision	Recall	Macro-F1
io	(tgt2src, 0.999)	0.705	0.240	0.359	0.649	0.239	0.350
io+ids	(tgt2src, 0.999)	0.835	0.347	0.490	0.797	0.346	0.483
io+cp	(src2tgt, 0.999)	0.917	0.750	0.825	0.895	0.748	0.815
io+ids+cp	(src2tgt, 0.999)	0.910	0.758	0.827	0.887	0.755	0.816
io+ids+cp (ex)	(src2tgt, 0.999)	0.896	0.771	0.829	0.869	0.771	0.817
io+ids+cp (ex+rp)	(src2tgt, 0.999)	0.905	0.771	**0.833**	0.881	0.771	0.822
io+co	(src2tgt, 0.997)	NA	NA	NA	0.937	0.564	0.704
io+co+ids	(src2tgt, 0.999)	NA	NA	NA	0.850	0.714	0.776
io+co+cp	(src2tgt, 0.999)	NA	NA	NA	0.880	0.779	0.826
io+co+ids+cp	(src2tgt, 0.999)	NA	NA	NA	0.899	0.774	0.832
io+co+ids+cp (ex)	(src2tgt, 0.999)	NA	NA	NA	0.882	0.787	0.832
io+co+ids+cp (ex+rp)	(src2tgt, 0.999)	NA	NA	NA	0.892	0.786	**0.836**
string-match	(combined, 1.000)	0.987	0.194	0.324	0.983	0.192	0.321
edit-similarity	(combined, 0.920)	0.971	0.209	0.343	0.963	0.208	0.343
LogMapLt	NA	0.965	0.206	0.339	0.956	0.204	0.336
LogMap	NA	0.935	0.685	0.791	0.918	0.681	0.782
AML	NA	0.892	0.757	0.819	0.865	0.754	0.806
LogMap-ML*	NA	0.944	0.205	0.337	0.928	0.208	0.340

Results on the FMA-SNOMED task.

Experiment: Key Observations

- Ablations of **BERTMap** settings:
  - *Unsupervised* settings can already perform rather well if input ontologies have sufficient class labels;
  - *Semi-supervised* settings are generally better than unsupervised ones.
  - The *complementary corpus* is extremely useful when input ontologies are deficient in class labels.
  - Mapping *extension* and *repair* consistently boost the performance.

- Comparisons to baseline models:
  - **BERTMap** is better than **LogMap** and **AML** on *FMA-SNOMED* and *FMA-SNOMED+* and slightly worse on *FMA-NCI*.
  - **BERTMap** is consistently better than the rest of the baselines, including both the string similarity-based symbolic models, and the machine learning model **LogMap-ML**.

Conclusions

- **BERTMap** mainly relies on what the input ontologies can provide (*unsupervised*) and can be further improved by given mappings (*semi-supervised*) and/or complementary resources (*data augmentation*).
- **BERTMap** considers the *text-level* (in mapping prediction), *graph-level* (in mapping extension), and *logic-level* (in mapping repair) contexts.
- **BERTMap** adopts a *simple but effective* candidate selection algorithm, where we use the *sub-word inverted indices* rather than the *word-level* ones which *(i)* require extra text processing and *(ii)* cannot deal with unknown words.

Future Work

- Benchmarking
  - To conduct more extensive evaluation on *large-scale data*.
  - To consider different application scenarios by using *different evaluation metrics* and *human evaluation*.
- Model
  - To incorporate *more than just equivalence mappings*.
  - To develop a more comprehensive and compact model with all sorts of contexts (*text-level, graph-level, logic-level)* collectively considered rather than processed separately.

Thank You!

BERTMap

- https://github.com/KRR-Oxford/BERTMap
- Codes & data for this paper.

DeepOnto

- https://github.com/KRR-Oxford/DeepOnto
- A developing python package for ontology engineering and evaluation.
- BERTMap will be re-implemented as an example model in this package.

This work is funded by Samsung Research UK (SRUK).