Università degli studi di Milano Bicocca
Department of Informatics, Systems and Communication

Master's degree program in Data Science

A systematic analysis of filter-based strategies for Entity Retrieval in tabular data

Supervisor: *Matteo Palmonari*
Co-supervisor: *Federico Belotti*

Master's degree thesis by:
Alessandro Belotti (ID 896985)

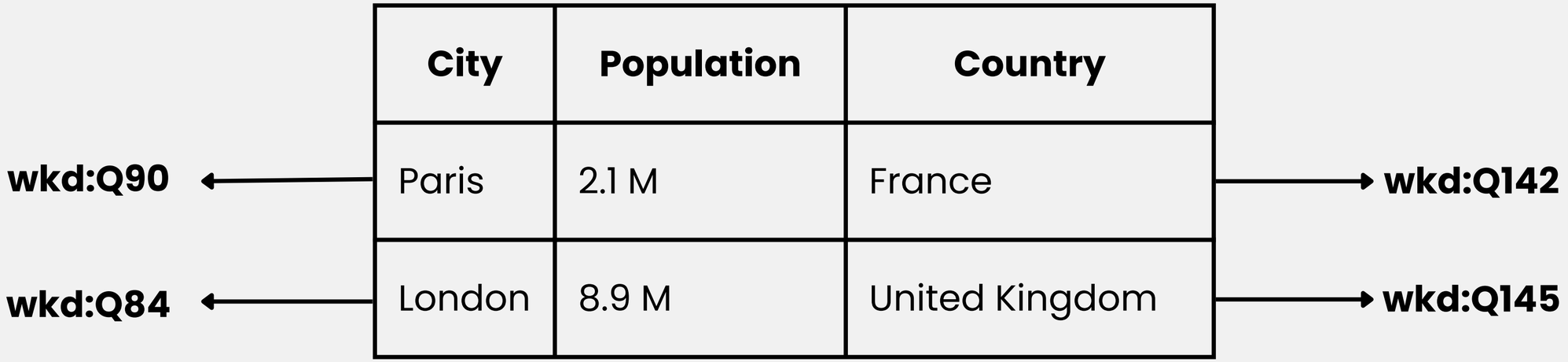
Academic Year 2023 - 2024

Contents

- Type-based filtering for Entity Linking in tabular data
  - Summary of contributions
- Type enrichment strategies
- Type-based filtering strategies
- Experiments setup and evaluation
- Conclusion
- Future works

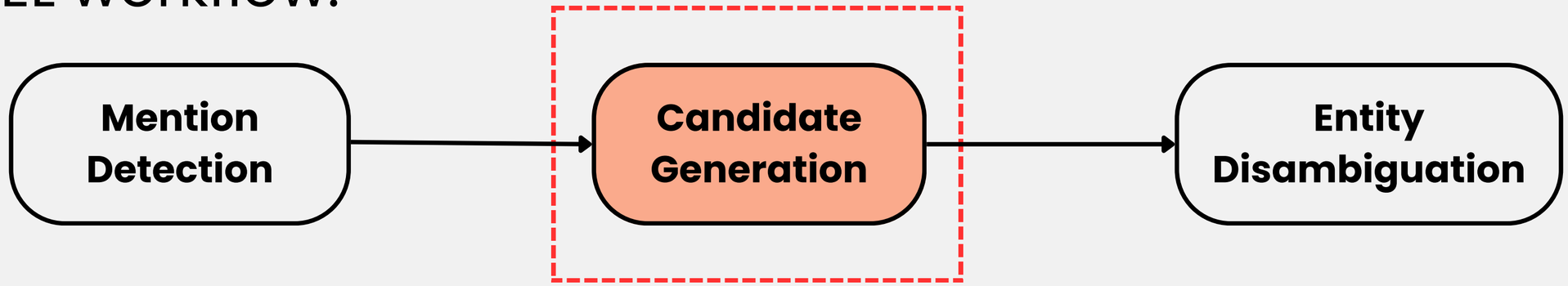
Type-based filtering for Entity Linking in tabular data

EL in tabular data has the goal of linking named entities mentioned in the table to their corresponding entities in **Wikidata** Knowledge graph.



Type-based filtering for Entity Linking in tabular data

The EL workflow:



LamAPI is the search engine used for the candidate generation. It's based on:

- **MongoDB** for the storaging
- **Elasticsearch** for indexing and querying the candidates

LamAPI
Label Matching API

Type-based filtering for Entity Linking in tabular data

Using **LamAPI** retrieval system with a simple label matching strategy there are some problems:

- many **irrelevant entities** included
- the correct candidate may be **excluded**

Entity Name	Entity ID	Entity Types
Paris Paris	Q111210960	television series
Paris Paris	Q42291481	film
Paris Paris	Q113859728	television series
Paris Paris Dix-Sept	Q106582858	municipal newsletter
Papilio paris paris	Q22104161	taxon
PARIS	Q124817316	scholarly article
Paris	Q2249942	Wikimedia set index article
Paris	Q974043	locality
Paris	Q121046410	song
Paris	Q19469915	dictionary entry
Paris	Q12503205	village in Indonesia
Paris	Q21094569	biographical article
Paris	Q7137193	metro station
Paris	Q786333	null
PARIS	Q64162970	photograph
Paris	Q28337101	musical work/composition
Paris	Q58733617	photograph
Paris	Q19222388	version, edition or translation

Type-based filtering for Entity Linking in tabular data

Discriminating entities by including only **geographic locations** helps filter out irrelevant entities and select the correct candidates.

Challenges include:

- **Disorder** across thousands of type hierarchies.
- **Low-quality** entity relationships.

Entity Name	Entity ID	Entity Types
Paris	Q974043	locality
Paris	Q12503205	village in Indonesia
Paris	Q7137193	unincorporated community in the United States
Paris	Q6065229	Corregimientos of Panama
Paris	Q3560147	constituency of the French Fifth Republic
Paris	Q2220917	civil town of Wisconsin
Paris	Q44873932	movie theater, arts centre
Paris	Q22134091	unincorporated community in the United States
Paris	Q2863958	arrondissement of France
Paris	Q3181341	city in the United States, county seat
Paris	Q960025	city in the United States
Paris	Q90	commune of France, department of France, capital city, metropolis, tourist destination, ...
Paris	Q7137161	unincorporated community in the United States
Paris	Q7137175	unincorporated community in the United States, census-designated place in the United States
Paris	Q25907009	kampung of Papua
Paris	Q576584	city in the United States
Paris	Q6922657	mountain
Paris	Q110940212	human settlement
Paris	Q79917	city in the United States
Paris	Q30621726	unincorporated community in the United States

Summary of contributions

The principal innovations in this thesis are:

1. Structured methodology for enhancing the **direct type** hierarchy in Wikidata
2. Integration of type-aware retrieval into **lamAPI** retrieval system
3. **Evaluation** of type-based filtering on retrieval performance:
    a. Impact across different tabular data domains
    b. Effect of varying candidate set sizes on retrieval metrics

Type enrichment strategies

Applying a type filtering strategy requires the definition of:

- a set called **query type**, in the context of tabular data given from the types associated to a mention
- a set of **candidate type**, given from the types associated to the entities in the KG

The domain of these types is defined as:

- **explicit WD type** (explicitly associated with an entity in Wikidata)
- **extended WD type** (extended from an explicit type through the *transitive closure*)
- **NER type** (maps the Wikidata entity explicit types to a generic type in a flat classification scheme)

Type enrichment strategies

Enrichment type strategies are a key focus in current scientific literature, but their implementation faces **challenges** due to the **disorganization** within the Wikidata class hierarchy [1].

To address this, state-of-the-art methods were selected and efficiently integrated into LamAPI to extend the index, considering the computational demands of processing millions of Wikidata entities.

The NER type methodology builds on an existing approach [2,3].

[1] Patel-Schneider, Peter F. et al (2024) Class Order Disorder in Wikidata and First Fixes

[2] Geiß, J., Spitz, A., & Gertz, M. (2018). Neckar: A named entity classifier for wikidata.

[3] A. L. F. Shanaz and R. G. Ragel (2019), “Named entity extraction of wikidata items,”

Type enrichment strategies

NER type enrichment

Having a list of explicit WD types mapped into a general class is possible to associate a **NER type** to an entity. The methodology is the following:

1. SPARQL query to Wikidata retrieves all the **subclasses** of:
   - *geographic location (Q2221906)* for LOCATION
   - *organization (Q43229)* for ORGANIZATION
   - *human (Q5)* for PERSON

Type enrichment strategies

NER type enrichment

2. Each entity's explicit type is checked against generated lists to determine its NER type. If no match is found, it’s mapped to the **OTHERS** type.
The final NER type for the entity is then defined based on this mapping.

Explicit WD types name	Explicit WD types id	NER type
sovereign state	Q3624078	LOC
federation	Q43702	ORG
country	Q6256	LOC
realm	Q1250464	LOC
member state of the European Union	Q185441	LOC
colonial power	Q20181813	ORG

Belgium (Q31) entity will have [ORG, LOC] as NER type.

Type enrichment strategies

Extended type enrichment

For each explicit entity type, the transitive closure of the Wikidata hierarchy is returned.

The union of all the types extension of an entity is its extended type.

Q6256*: *country, *Q1048835: political territorial entity, Q4835091: territory, Q56061: administrative territorial entity, Q82794: geographic region, Q15642541: humangeographic territorial entity, Q16562419: political entity, Q618123: geographical feature, Q26713767: region of space, … Q27096213: geographic location, …*

Type enrichment strategies

The workflow used integrating the methodologies in LamAPI is the following:

1. **Storaging** of the Wikidata dump file into MongoDB, included for each processed entity
   a. *WD types* of the entity
   b. *NER types* mapping
   c. *extended WD types*

2. **Indexing** of the entities in an Elasticsearch index
   The indexed fields serve as *searchable* attributes to efficiently find relevant candidates.

Type-based filtering strategies

The filtering operation can be defined as a combination of these inputs:

- mention label
- query type set
- constraint mode:
  - **hard** (if all the candidate types are not matching the query type the candidate is filtered out)
  - **soft** (all the candidates are included but the score is updated based on the candidate types that are matching the query types)

The output is a set of **candidates** where each candidate $c_i \in C$ is defined as the triple:

$$c_i = \langle \mathrm{id}_i, T_i, s_i \rangle$$

Type-based filtering strategies

The original label-matching strategy has been extended:

1. exact match between mention and candidate labels
2. error tolerance provided by **fuzzy search**
3. entity types are used as filterable attributes in hard or soft mode

Example of query with type-based filtering and a **hard** constraint with fuzzy search.

```
{
    "query": {
        "bool": {
            "must": [
                {"match": {"name": {"query": "Paris", "boost": 2.0, "fuzzness":"AUTO"}
                { "terms": { "extended_WDtype": [ "Q5119", "Q484170" ] } }
            ]
        }
    }
}
```

Example of query with type-based filtering and a **soft** constraint with fuzzy search.

```
{
    "query": {
        "bool": {
            "must": [{"match": {"name": {"query": "Paris", "boost": 2.0,
            "fuzzness":"AUTO"}}}],
            "should": [
                { "term": { "extended_WDtype": "Q5119" } },
                { "term": { "extended_WDtype": "Q484170" } }
            ]
        }
    }
}
```

Type-based filtering strategies

The combinations of different type domains for query and candidate type set create multiple type filtering strategies:

		Candidate types		
		explicit_WDtype	NERtype	extended_WDtype
Query types	explicit_WDtype			
	NERtype			
	explicit_WDtype			

The green cells are referred to strategies where the query type and candidate type are joint sets, so they are feasible.
Red cells are unfeasible strategies and the orange cell is the approach without enrichment.

Experiments setup and evaluation

There goals of the experiments are:

- evaluate if the correct candidate it’s included in the first k=100 positions (**coverage**)
- assess the **average position** in the top-k 100 candidates (**MRR**)
- understand how the tabular data affects the previous experiments
- impact of the **candidate set size** on the experiments

Experiments setup and evaluation

The experiments evaluate strategies are based on the following combinations of **query type** sets and **candidate type** sets:

- *[NER type - NER type]*
- *[Explicit WD type - Extended WD type]*
- *[NER type - Extended WD type]*

The evaluation metrics used are coverage and MRR :

$$\text{coverage} = \frac{\#\text{ candidates found}}{\#\text{ total candidates to find}}$$

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i}$$

Experiments setup and evaluation

General domain dataset

Query Type	Candidate Type	Mode	Round1	Round3	Round4	2T	HardTable2	HardTable3
no filter			0.827	0.890	0.823	0.540	0.884	0.741
NERtype	NERtype	soft	0.827	0.892	0.801	0.544	0.880	0.750
NERtype	NERtype	hard	0.789	0.770	0.698	0.534	0.744	0.542
explicit_WDtypes	extended_WDtypes	soft	0.827	**0.899**	**0.869**	**0.550**	**0.914**	**0.826**
explicit_WDtypes	extended_WDtypes	hard	0.636	0.752	0.789	0.525	0.855	0.781
NERtype	extended_WDtypes	soft	**0.846**	0.859	0.804	0.544	0.868	0.755
NERtype	extended_WDtypes	hard	0.731	0.618	0.720	0.539	0.768	0.617

Coverage results at N = 100

Query Type	Candidate Type	Mode	Round1	Round3	Round4	2T	HardTable2	HardTable3
no filter			0.797	0.807	0.861	0.435	0.858	0.800
NERtype	NERtype	soft	0.815	0.816	0.878	0.468	0.867	0.832
NERtype	NERtype	hard	0.776	0.708	0.712	0.451	0.716	0.556
explicit_WDtypes	extended_WDtypes	soft	**0.856**	**0.826**	**0.913**	**0.481**	**0.900**	**0.909**
explicit_WDtypes	extended_WDtypes	hard	0.842	0.815	0.707	0.461	0.726	0.600
NERtype	extended_WDtypes	soft	0.821	0.824	0.868	0.474	0.881	0.834
NERtype	extended_WDtypes	hard	0.712	0.565	0.762	0.469	0.744	0.663

MRR results at N = 100

- **Soft filters** outperform hard filters in coverage and MRR, especially when filtering by:
  - Explicit WD type vs. Extended WD type
- **HardTableR3** generally has the highest difference between soft and hard strategies.
- **2T** is the most challenging dataset, with the worst performance.

Experiments setup and evaluation

Domain-specific dataset

Label	Wikidata ID
The Graduate Center, CUNY	Q1024543
Zhejiang University \& Westlake University	Q986087
BAAI	Q107518033
Capgemini Spa	Q1034621
Alation	Q107639776

Example of labels with the correct entities id.

Query Type	Candidate Type	Mode	Coverage	MRR
no filter			0.712	0.680
NERtype	NERtype	soft	0.722	0.707
NERtype	NERtype	hard	0.714	0.666
NERtype	extended_WDtype	soft	**0.727**	**0.709**
NERtype	extended_WDtype	hard	0.709	0.664

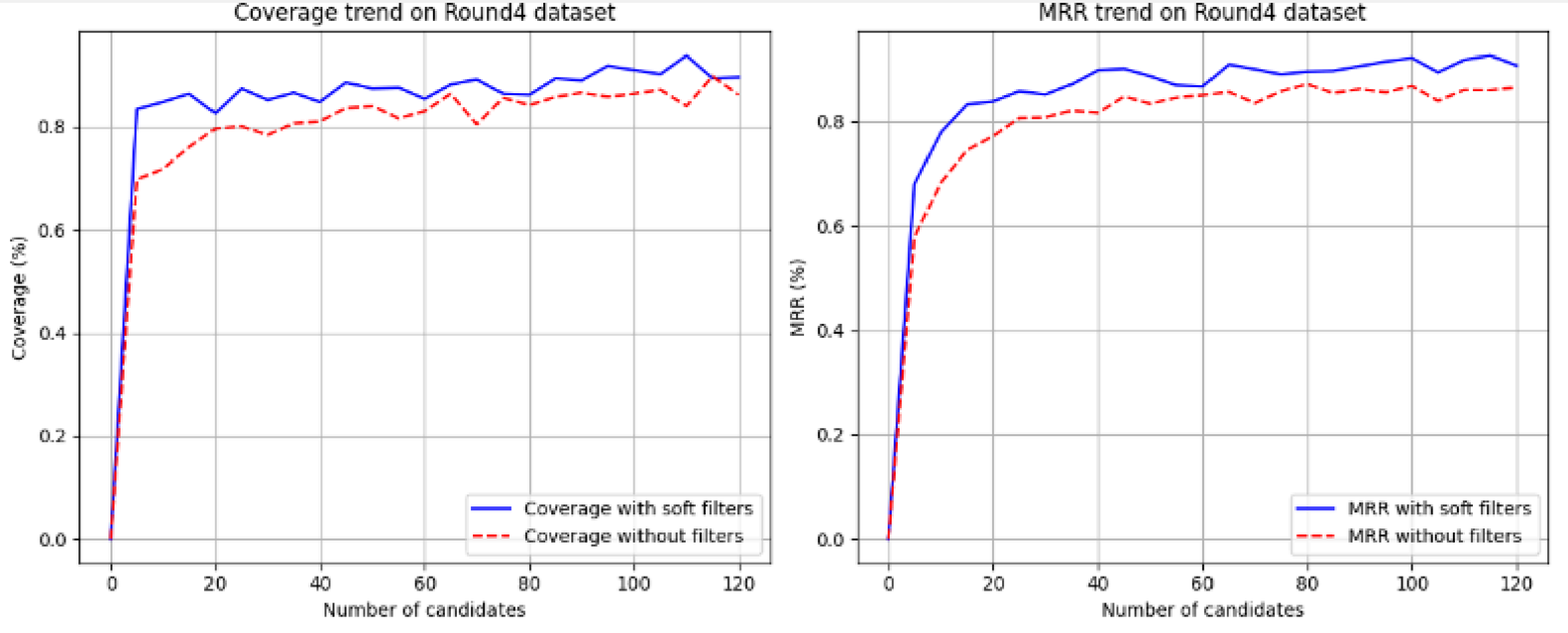
Results at N = 100

Entity type filters help in interactive linking within vertical domains by eliminating irrelevant candidates based on **known entity types**.

Here the reason of evaluating a dataset composed only by *organizations.*

The best strategy is still **soft constraint** with NER type and extended type.

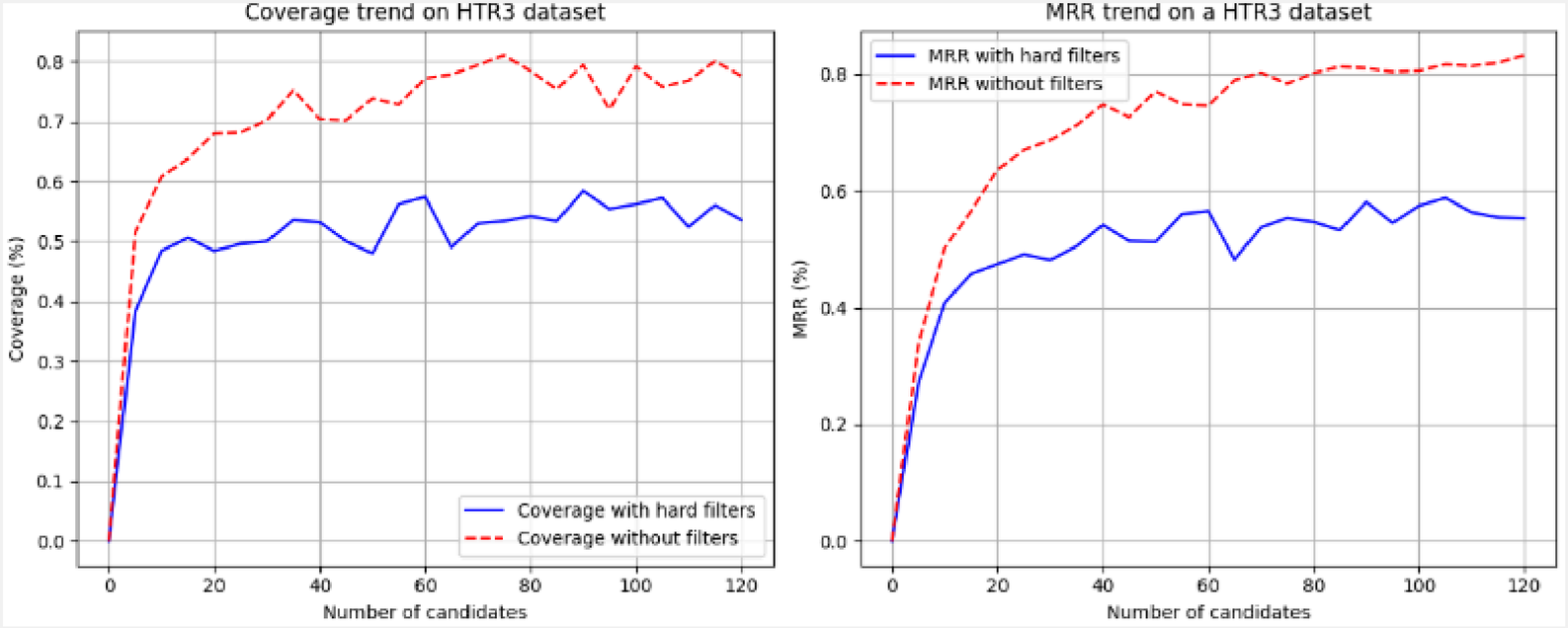
Experiments setup and evaluation



Coverage and MRR trends for Round4 with *[Explicit WD type - Extended WD type]* **soft** strategy.

Round4 outperforms other datasets, with coverage and MRR trends showing the soft strategy's effectiveness, regardless of candidate set size.

Experiments setup and evaluation



Coverage and MRR trends for HardTableR3 with *[NER type - NER type]* **hard** strategy.

HardTableR3 shows the lowest coverage and MRR in hard mode, making it the choice for this experiment. Insights reveal that hard filters do not outperform the no-filter strategy, regardless of candidate set size.

Experiments setup and evaluation

HardTableR3 shows the lowest coverage and MRR in hard mode, linked to label spelling quality, not candidate set size.
In soft mode, it boosts candidate scores, including the correct one.

For example, with the [NER type - NER type] strategy, the correct candidate **Independencia** for the mention *Indepepdencia* is retrieved only using a soft constraint:

Candidate set using a **soft** constraint [NER type - NER type] strategy

Position	ID	Name	Description	Score
1	Q22020934	Independencia, Independencia	Settlement in Independencia, Ceara, Brazil	1.000
2	Q20142558	San José Independencia	Human settlement in Mexico	0.678
3	Q20230272	Independencia	Settlement in Puebla, Mexico	0.651
4	Q987035	Independencia	Hamlet in Florida, Uruguay	0.651
5	Q790471	Independencia	Street in Buenos Aires, Argentina	0.650
...	...	...	...	...
53	Q1190620	Independencia	Department of Argentina	**0. 590**
...	...	...	...	...
99	Q5921669	Independencia Island	Island in Peru	0.581
100	Q5716566	Bahía Independencia	Bay in Peru	0.581

Candidate set using a **hard** constraint [NER type - NER type] strategy

Position	ID	Name	Description	Score
1	Q20230272	Independencia	Settlement in Puebla, Mexico	1.000
2	Q987035	Independencia	Hamlet in Florida, Uruguay	1.000
3	Q790471	Independencia	Street in Buenos Aires, Argentina	0.998
4	Q1576604	Independencia	District of Huaraz, Peru	0.998
...	...	...	...	...
98	Q27774284	Independencia	Human settlement in Bácum, Sonora, Mexico	0.997
99	Q106230310	INDEPENDENCIA 256	Archaeological site in Argentina	0.886
100	Q3150018	Independencia Parish	Municipio Libertador, Carabobo State, Venezuela	0.884

Experiments setup and evaluation

When mention labels contain a high number of errors, as in the **2T** tabular dataset, neither hard nor soft constraints effectively improve retrieval performance.

It containes mispelled mentions like **Steve Blackkk** for the entity *Steven Black* (Q7614497) or **Johnh Callahannn** for J*ohn Callahan* (Q1699525).

Experiments setup and evaluation

- The **soft type-based strategy** outperforms baseline label matching in LamAPI, especially with the [Explicit WD type - Extended WD type] strategy.

- This holds true even for **multi-token mentions**, provided misspellings are minimal.

- Type-based filters do not help much when the ranking is too noisy (in this case, **injected misspelling**).

Conclusion

- Defined and systematically evaluated different type-based enrichment strategies for type-based filtering across various settings.
- Extended a retriever tool to incorporate these strategies.
- Demonstrated that one of these strategies outperforms the baselines and other alternatives.

Future works

- Improve label-matching in LamAPI to **handle misspellings** and noisy data
- Enhance type extension to better **handle the disorder** in the Wikidata class hierarchy.
- Evaluate EL impact, studying **retrieval impact** on the disambiguation phase