Evaluating the Use of Synthetic Queries for Pre-Training a Semantic Query Tagger

Elias Bassani[1,2] and Gabriella Pasi[2]

[1]Consorzio per il Trasferimento Tecnologico - C2T [2]University of Milano-Bicocca, Milan, Italy

What is Semantic Query Labeling?

- **Semantic Query Labeling** is the task of **locating** the constituent parts of a query (*segmentation*) and **assigning** domain-specific semantic labels to each of them (*classification*).
- It **unfolds** the **relations** between the **query terms** and the **documents' structure** while leaving unaltered the keyword-based query formulation.
- **Example:** "alien ridley scott 1979"
  1. **Segmentation:** "alien" "ridley scott" "1979"
  2. **Classification:** "alien" → *#title* "ridley scott" → *#director* "1979" → *#year*

Summary

- We investigate the **pre-training** of a BERT-based **semantic query-tagger** with **synthetic data** generated by leveraging the documents' structure.
- By simulating a **dynamic environment**, we also evaluate the **consistency** of performance improvements brought by pre-training as real-world training data becomes available.
- The **results** of our experiments **suggest** both the **utility of pre-training** with synthetic data and its improvements' **consistency over time**.

Research Questions

1. Can we improve the performance of a semantic query tagger by pre-training it with *synthetic* data before fine-tuning it with *real-world* queries?
2. Can pre-training with many synthetic queries solve the inconsistency of a model in predicting semantic classes under-represented in the training set?
3. Is the performance boost given by pre-training, if any, consistent over time while new real-world training data become available?
4. When does fine-tuning with real-world data become effective for achieving performance improvements over a model trained only on synthetic queries?

Dataset

- Structured movie-related document collection from Kaggle.
- Movie-related queries extracted from the AOL query logs and manually tagged using the following semantic labels: *Title*, *Country*, *Year*, *Genre*, *Director*, *Actor*, *Production*, *company*, *Tag* (mainly topics and plot features), *Sort* (e.g., new, best, popular, etc.)
- Three evaluation scenarios of increasing difficulty: *Basic*, *Advanced*, *Hard*.

Table 1. Statistics of the benchmark datasets.

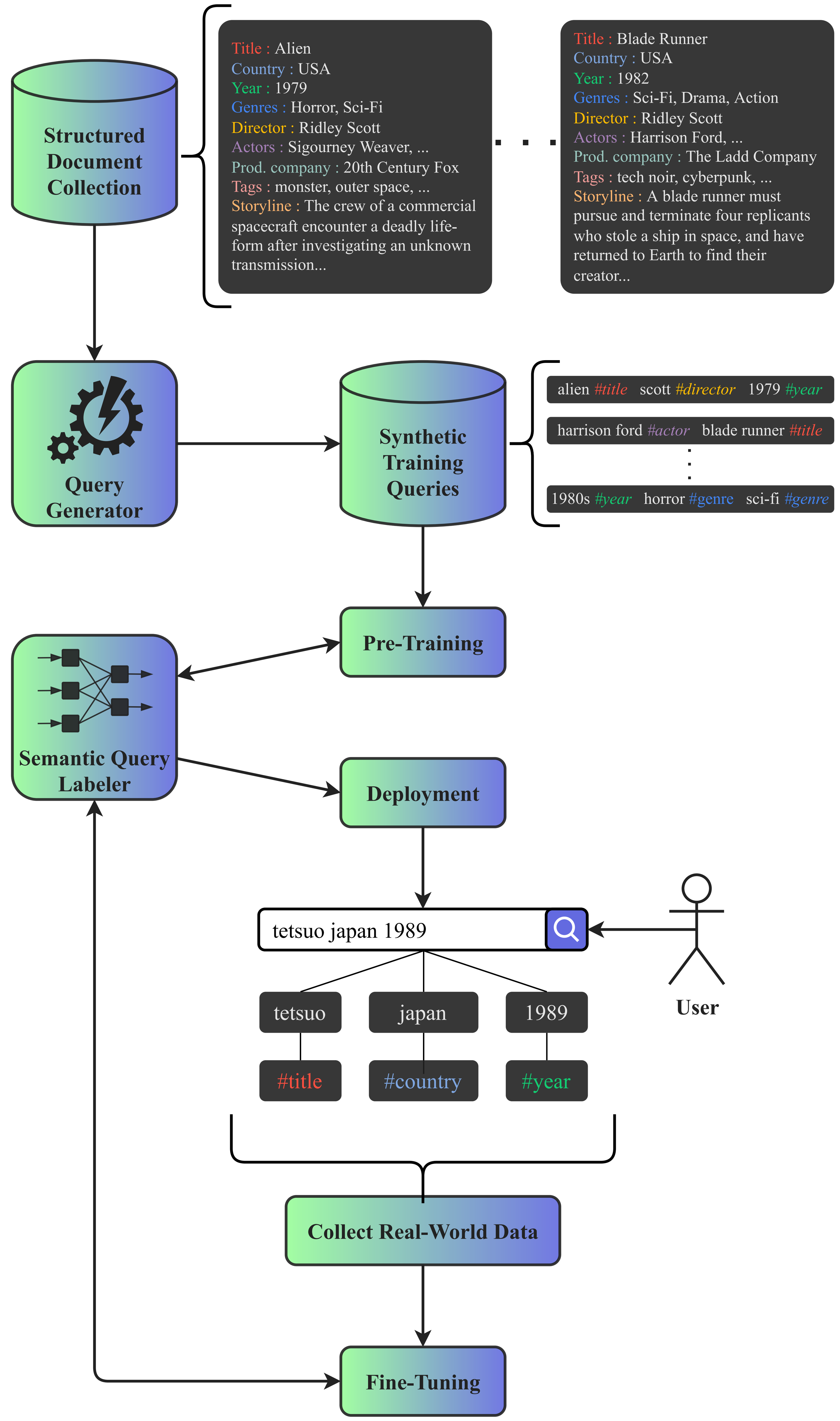
	Basic	Advanced	Hard
# train queries	3938	4292	5131
# dev queries	601	672	822
# test queries	538	610	796
Total	5077	5574	6749

Figure 1. More about the dataset

Compared Models

1. **Synthetic:** model trained with 100k synthetic queries.
2. **Real:** model trained with queries from the real-world dataset.
3. **Pre-trained:** model pre-trained on synthetic data and fine-tuned with the real-world queries.

Pipeline



Experiment 1

- **Goal:** Evaluate the performance gains we can achieve by pre-training a semantic query tagger with synthetic data generated from a structured corpus and fine-tuning it with real-world queries.
- **Findings:**
  1. *Pre-trained* model consistently outperforms the considered baselines in all the evaluation scenarios.
  2. Most noticeable benefits of pre-training / fine-tuning on *Hard*, the most complex of the considered scenarios.
  3. Synthetically generated queries can play a complementary role w.r.t. real-world queries in effectively training a semantic query tagger — by pre-training a semantic query tagger with many synthetic queries, we can expose the model to abundant in-domain and task-related information.

Table 2. Overall effectiveness of the models. Best results are in boldface.

Model	*Basic*		*Advanced*		*Hard*	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Synthetic	0.909	0.884	0.903	0.865	0.765	0.756
Real	0.927	0.903	0.896	0.776	0.816	0.756
Pre-trained	**0.934**	**0.910**	**0.925**	**0.893**	**0.840**	**0.828**

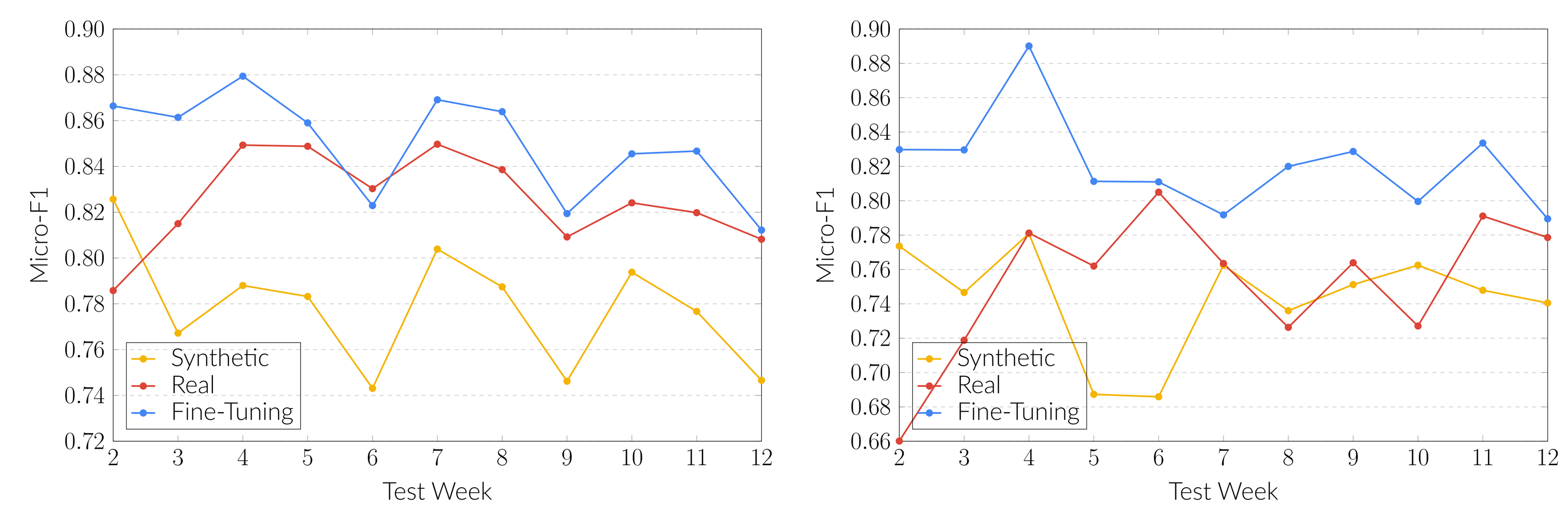
Table 3. F1 scores for each model and semantic class. Best results are in boldface.

Scenario	Model	*Actor* F1	*Country* F1	*Genre* F1	*Title* F1	*Year* F1	*Director* F1	*Sort* F1	*Tag* F1	*Company* F1
Basic	Synthetic	0.898	0.811	0.867	0.917	0.928	*N/A*	*N/A*	*N/A*	*N/A*
Basic	Real	0.865	**0.857**	**0.897**	**0.949**	0.945	*N/A*	*N/A*	*N/A*	*N/A*
Basic	Pre-trained	**0.905**	**0.857**	0.862	0.945	**0.978**	*N/A*	*N/A*	*N/A*	*N/A*
Advanced	Synthetic	0.885	0.833	**0.923**	0.914	0.983	0.667	0.853	*N/A*	*N/A*
Advanced	Real	0.844	0.765	0.880	0.921	0.975	0.111	**0.937**	*N/A*	*N/A*
Advanced	Pre-trained	**0.890**	**0.849**	0.895	**0.937**	**1.000**	**0.750**	0.929	*N/A*	*N/A*
Hard	Synthetic	0.857	0.773	0.855	0.777	0.971	0.550	0.876	0.522	0.623
Hard	Real	0.831	**0.837**	0.873	0.854	0.956	0.222	0.883	0.576	0.771
Hard	Pre-trained	**0.884**	0.809	**0.897**	**0.857**	**0.985**	**0.667**	**0.931**	**0.600**	**0.817**

Experiment 2

- **Setting:** Simulation of a dynamic environment — based on real-world data — where new labeled queries are collected over time.
- **Goal:** Evaluate the consistency over time of the improvements brought by pre-training with synthetic data.
- **Findings:**
  1. Improvements brought by pre-training are consistent over time.
  2. As soon as we fine-tune with real-world queries the model pre-trained on synthetic data, it achieves top performances.
  3. While we collect real-world training data for conducting fine-tuning, we can employ *Synthetic* with no actual performance loss w.r.t. *Real*.

Figure 2. Over time effectiveness of the models in the HARD scenario.



(a) Micro-F1 scores over time in HARD scenario (b) Macro-F1 scores over time in HARD scenario