CitySAT: a system for the semantic answer type prediction

MSc Student : Chaeyoon Kim, Supervisor: Dr Ernesto Jimenez-Ruiz @ City, University of London

Research Aim:

- This study starts from participation in the **Se**Mantic **A**nswer Type and Relation Prediction **T**ask (SMART) 2021 **[1]** whose task aims at predicting the answer type for the given natural language questions.

Initial Investigation

- There are four attribute-value pairs in JavaScript Object Notation (JSON) format.
- Evaluation Matrix: Accuracy, Normalized Discounted Cumulative Gain (NDCG) @5, @10

Table 1. SMART AT Task Data Characteristics: Overall data size increased 231%↑ for 2021 edition.

2020 version			2021 version		
Train	Test	Total	Train	Test	Total
(6 May 2020)	(9 Sep 2020)		(15 July 2021)	(14 Sep 2021)	
17,571	4,369	21,940	40,621	10,093	50,714

{
"id": "1",
"question": "Who are the gymnasts coached by Amanda Reddin?",
"category": "resource",
"type": ["dbo:Gymnast", "dbo:Athlete", "dbo:Person", "dbo:Agent"]
}

{
"id": "1",
"question": "Who are the gymnasts coached by Amanda Reddin?"
}

[Left] Training data [Right] Test data in JSON format

Baseline analysis

- In 2020, the comparable eight systems accuracy are mostly quite high over 90% **[2]** meaning that CPU processing may be sufficient to complete our research models.
- For the 2021 dataset, an overview of runtime estimation(~30 mins) and answer category classification model accuracy(0.87, 0.88, 0.85) has verified for SVM, LR, MLP respectively on Google Colab.

Question Sentence Parsing

- Used a Sci-kit learn library: *feature_extraction.text.CountVectorizer(TF) and TfidfVectorizer*

Text Normalization and Hypothesis Verification

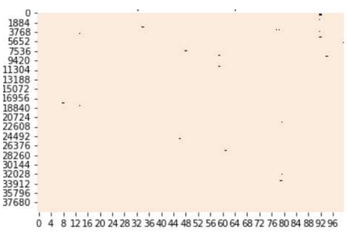
- Deployed a python program for tokenization, stemming and lemmatization (lemma) on Google Colab environment using the Natural Language Toolkit (NLTK) library.
- For Wh-clauses, if our programmed stop words are activated, the word collection skips to pick the word when it is one of the stop words. However, it didn't help in performance.

Combination	Accuracy	NDCG@5	NDCG@10
1 (no stop words, lemma and stemming order)	0.973	0.735	0.656
2 (no stop words, stemming and lemma order)	0.973	0.739	0.660
3 (one-time stop words – upfront)	0.916	0.635	0.563
4 (twice time stop words – upfront and afterwards)	0.915	0.629	0.558
5 (one-time stop words – afterwards)	0.920	0.634	0.562

Table 3. Some of the Exploration with different combinations of text normalization.

Bag of Words (BOW) Vectorization

Figure 1. Density graph of the vectorizer result



- Used a Sci-kit learn library.
- The colour palate (Fig1) is representing the mapped parts and the black dots/lines are displaying discrete/continuous null space in mapping.
- Feature selection has been applied to pick affordable number, 10K of unigrams and bigrams from the tokenized and normalized sentences because of the limited CPU processing threads.

Answer Type Reframing

Figure 2. Tabular representation of the dataset and the count number of the unique answer types for each location

	type1	type2	type3	type4	type5	type6	type7	type8	type9	type10
0	boolean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	dbo:Opera	dbo:MusicalWork	dbo:Work	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	date	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
>>> the number of type1: 302
>>> the number of type2: 173
>>> the number of type3: 134
>>> the number of type4: 130
>>> the number of type5: 87
>>> the number of type6: 58
>>> the number of type7: 37
>>> the number of type8: 37
>>> the number of type9: 30
>>> the number of type10: 23
```

```
>>> array([nan, 'dbo:Location', 'dbo:Cleric', 'dbo:Group', 'dbo:Stream',
    'dbo:Region', 'dbo:MilitaryUnit', 'dbo:PopulatedPlace', 'dbo:Band',
    'dbo:Settlement', 'dbo:Village', 'dbo:Country',
    'dbo:AdministrativeRegion', 'dbo:River', 'dbo:City', 'dbo:Place',
    'dbo:Agent', 'dbo:MusicalArtist', 'dbo:MusicGenre', 'dbo:Station',
    'dbo:Town', 'dbo:NaturalPlace', 'dbo:RollerCoaster', 'dbo:Work',
    'dbo:Ship', 'dbo:CityDistrict', 'dbo:Mammal', 'dbo:BodyOfWater',
    'dbo:Airport', 'dbo:Broadcaster'], dtype=object)
```

- Since the type1 embraces all types of three categories, the pure ontology classes are 298 in the first location. We could see other every location includes the 'nan' as a unique value as above which is going to be deleted in our modelling progress.
- Therefore, the maximum number of types is 298 and the minimum is 22.

MLP					
2_type1	2_type2	2_type3	2_type4	...	2_type10
Ontology 1	Ontology 1	Ontology 1	Ontology 1	...	Ontology 1
...	...	...	...	...	...
298	172	133	129	...	22

Methodology:

1. **Data loading:** Create the tabular representation of the dataset.
2. **Data Manipulation:** Clean the data and merged two years dataset. (Final Training: 39,556 / Test: 9,104)
3. **Pre-processing:** 80% Training, 20% Validation.
   i. Question: Sentence Parsing, Text Normalization, BoW Vectorization
   ii. Answer: Type Reframing
4. **Prediction:** Stage1 Category (LR), Stage2 Type Literal (LR), Resource (MLP)
5. **Formatting:** Module Evaluation, Save System Output

Logistic Regression Classifier (LR)

- As a discriminative classifier, it directly model the likelihood P(Y|X) or f: X→Y in ascending order.
- Parameters retains a probabilistic semantics.
- Pros: A simple linear classifier with a few parameter learned by iterative. optimization, and suppose less assumption
- Cons: Slow speed at the beginning **[3]**.

Multi-Layer Perceptron (MLP)

- By the benefit of scikit-learn, input and output numbers automatically defined.
- Suitable to deal with various number of features (~760 classes for DBpedia)
- Pros: Adaptive learning. Easy to discover based on the data given for training or initial experience.
- Cons: Many justification and decision for parameters.

Exceptional Hyper-parameters for two LRs and a MLP model

- LG1: penalty = 'elasticnet', solver = 'saga', l1_ratio = 0.2, verbose = 2.
- LG2: l1_ratio = 0.5, others same as LG1.
- MLP: hidden_layer_sizes=(1000, 500, 300), verbose = 2.

Analysis and Evaluation of Results

Table 2. CitySAT Results table

Submission version	Validation			Test		
	Accuracy	NDCG@5	@10	Accuracy	NDCG@5	@10
1	0.969	0.732	0.649	0.970	0.778	0.683
2	0.973	0.735	0.656	0.981	0.839	0.739
3*	0.953	0.699	0.622	0.967	0.810	0.713
4	0.973	0.737	0.658	0.984	0.836	0.737
5	0.973	0.736	0.656	0.985	0.842	0.742
6 (✓ Best Results)	0.973	0.736	0.738	0.984	0.842	0.854

Table 3. Config settings of the best model

Version	Stopwords	Stemming	Lemma	Embedding	Iteration	Type
6 Best	FALSE	TRUE	FALSE	TF	200/10	10

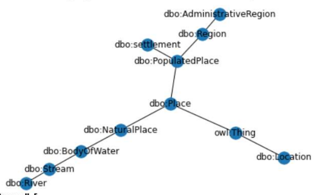
*In comparison with the prior submissions, it dropped ~2% of every evaluation score. This included the stopwords' changes and TF-IDF embeddings which mean the system used Wh-terms question inputs in TF-IDF embeddings

- **The Best Model results 0.984 / 0.842 / 0.854 on evaluation matrix (runtime ~45mins)**

@chaeyoonyunakim /smart-2021-AT_Answer_Type_Prediction/blob/main/SMART2021_AT_Prediction_Task_v6.ipynb

Limitations

"id":23678, "question":"on which river is paris"



"type":[
"dbo:Place",
"dbo:Location",
"dbo:BodyOfWater",
"dbo:NaturalPlace",
"dbo:River",
"dbo:Stream",
"dbo:AdministrativeRegion",
"dbo:Settlement"
]

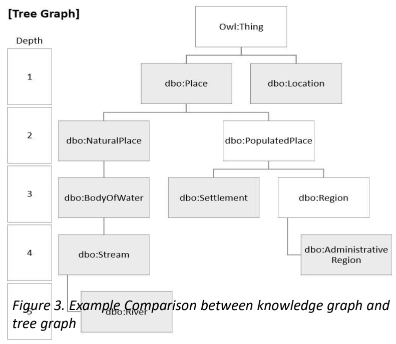


Figure 3. Example Comparison between knowledge graph and tree graph

- Disorder arrangement in answer types.
- Missing information for updated depth.

Lessons and Future work

- For later work, it would be worth to split training dataset into Knowledge based labels.
- Training the logistic regression model on the full dataset was time intensive and yielded complex coefficients. Either implementing a bespoke algorithm or using different software in the future for implementing the logistic model would be best.

References:

1) Mihindukulasooriya, N., Dubey, M., Gliozzo, A., Lehmann, J., Ngomo, A., Usbeck, R., Rossiello, G. and Kumar, U., 2021. SMART Task 2021 | SeMantic AnsweR Type and Relation Prediction Answer task. [online] Smart-task.github.io. Available at: <https://smart-task.github.io/2021/> [Accessed 1 October 2021].
2) CEUR-WS.org, 2020. SeMantic AnsweR Type prediction task (SMART) at ISWC 2020 Semantic Web Challenge (SMART). CEUR-WS, [online] Vol-2774. Available at: <http://ceur-ws.org/Vol-2774/> [Accessed 1 October 2021].
3) Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, pp.841-848.