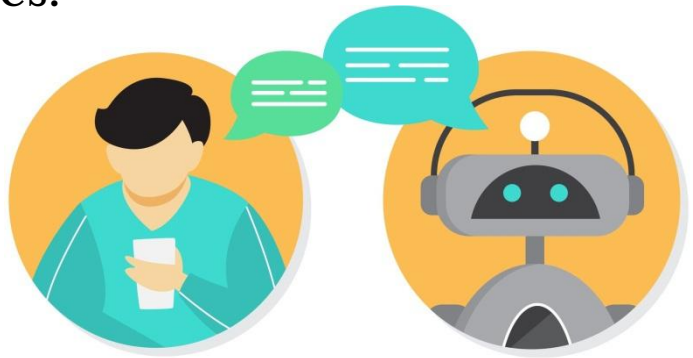
A Methodology for Hierarchical Classification of Semantic Answer Types of Questions

Ammar Ammar
Remzi Celebi
Shervin Mehryar

05-11-2020

Introduction

- Question answering systems have recently been integrated with many smart devices and search engines.



- Predicting the answer type in the question systems will help filter irrelevant results and improve overall search and retrieval performance.

- Predicting granular answer types for a question from an ontology is a greater challenge due to the large number of possible classes.

Questions categories

- Questions can be divided by their answer type into 3 categories:
  - Boolean answers questions:, with Yes/No answers.
  - Literal answers questions, with number/string/date answers.
  - Resource answers questions, with answer types that can branch into much granular types (following an ontology classes hierarchy for example) .

Problem statement

- Given a question in natural language, can we predict the categories and types of the given question from the WikiData and the DBpedia ontologies?

Methodology

- We divided the problem into two parts:
  - Predict the questions categories.
  - Predict the questions types.

Methodology

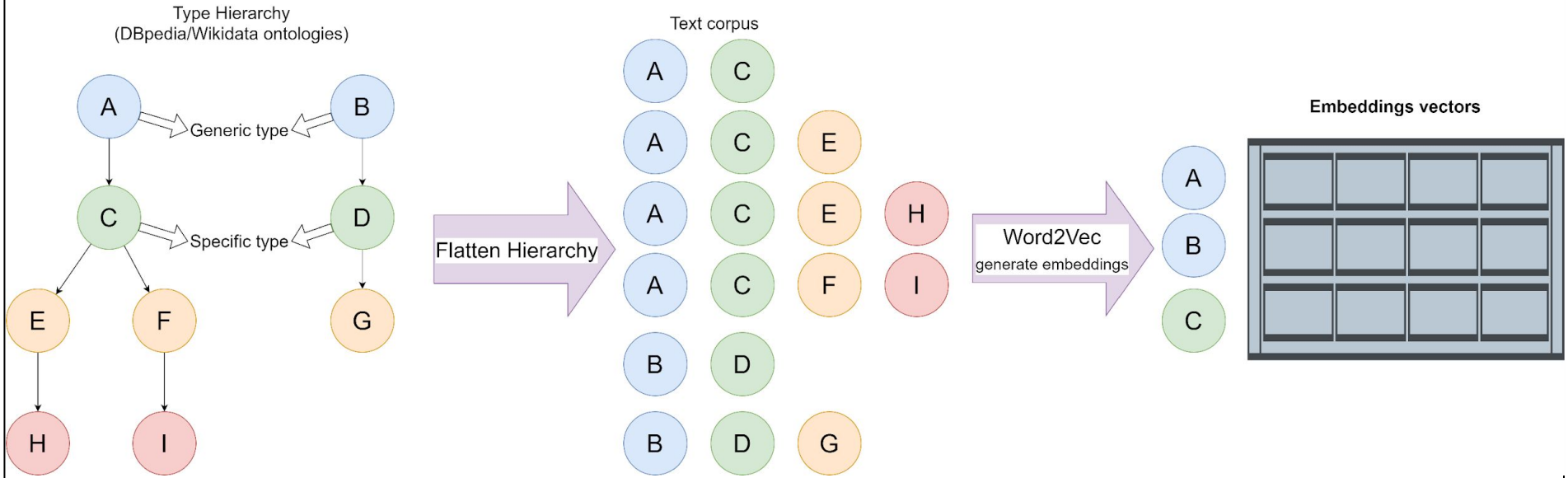
- Question category prediction:
  - The problem is modeled as multi-class classification to predict one of the extended categories (boolean, number, string, date and resource).
  - Fastai text classification library (Python) is used.
  - Tokenization and numericalization “if needed” is handled automatically within the fasiai.
  - The trained model is a deep learning LSTM neural network.
  - Obtained accuracy: 0.95

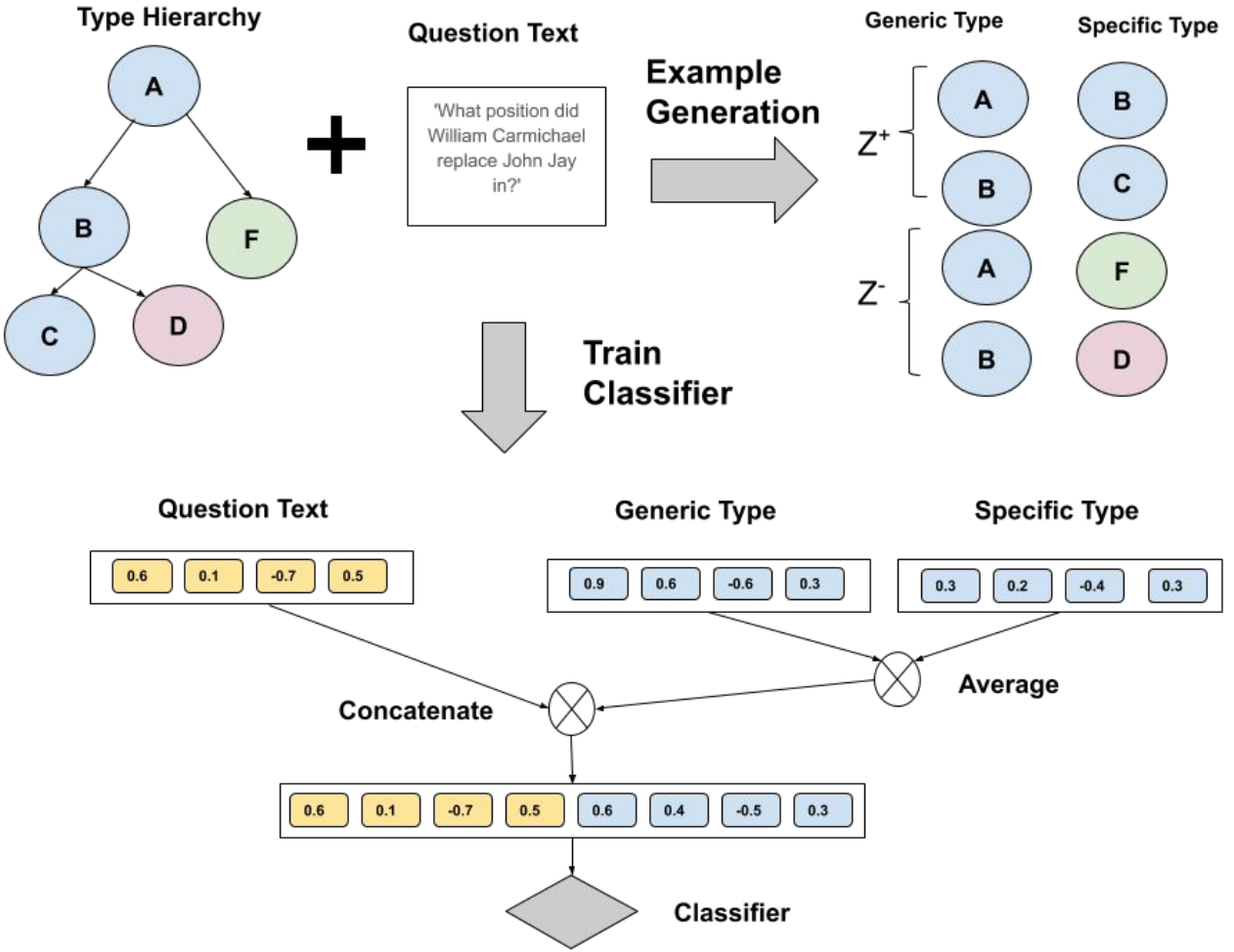
Methodology

- Question generic type prediction:
  - We first obtain the most frequent type among all the question types that are predicted to have a ‘resource’ category.
  - Predict the generic type from the question text using a text classification model (LSTM neural network) of the fastai library.
  - Obtained accuracy on validation data: 0.73

Methodology

- Question specific type prediction.




Type Hierarchy
A
B
F
C
D
Question Text
'What position did William Carmichael replace John Jay in?'
Example Generation
Generic Type
Specific Type
Z+
A
B
B
C
Z-
A
F
B
D
Train Classifier
Question Text
0.6
0.1
-0.7
0.5
Generic Type
0.9
0.6
-0.6
0.3
Specific Type
0.3
0.2
-0.4
0.3
Concatenate
Average
0.6
0.1
-0.7
0.5
0.6
0.4
-0.5
0.3
Classifier

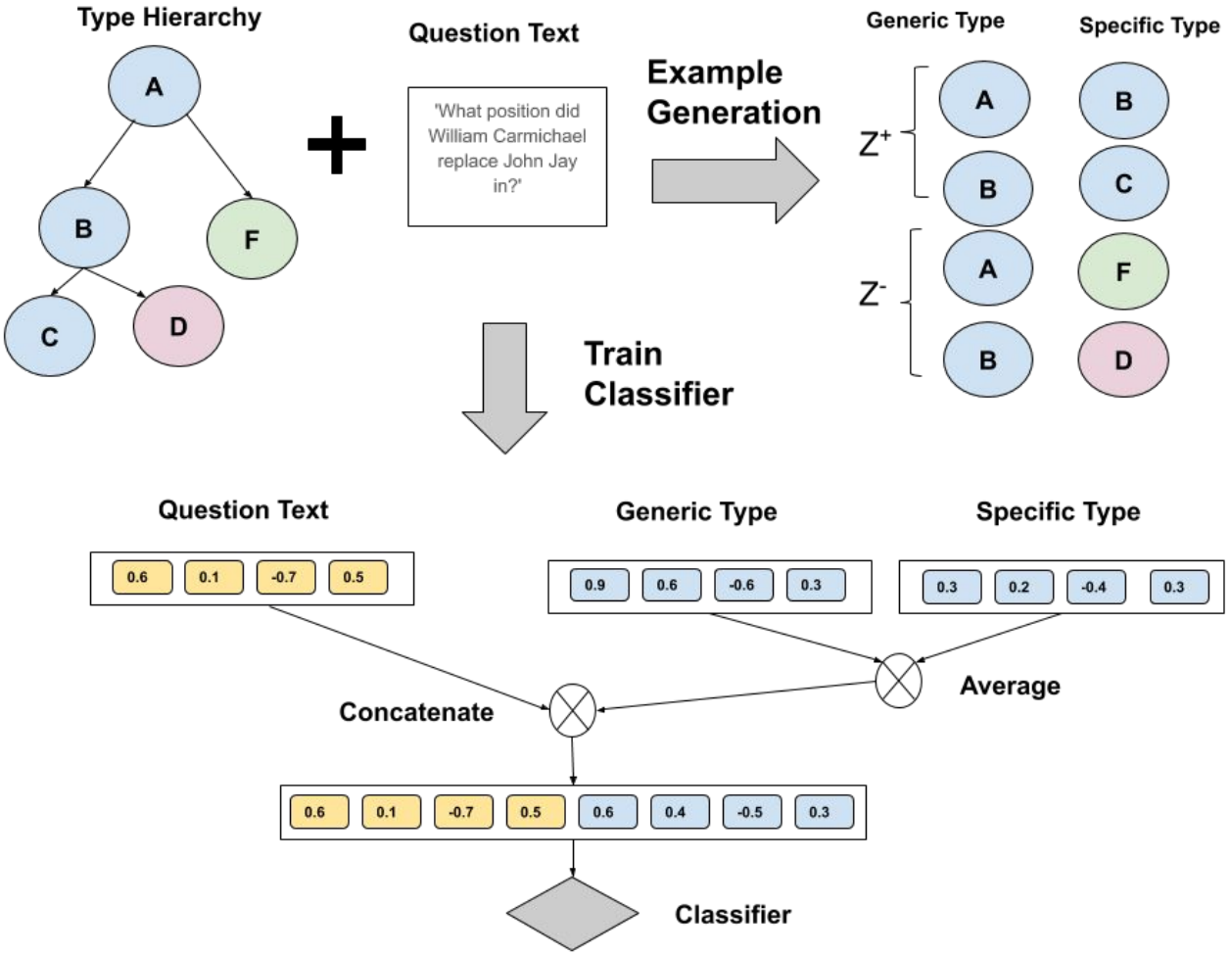
Methodology

- Question specific type prediction:
  - For the question text, word and entity embeddings were extracted using wikipedia2vec.
  - The embeddings of tokens (word or entity), extracted from each question, were averaged to obtain a fixed size vector for each question.
  - For generic and specific types, vector representation was learnt by training a word2vec model.

Methodology

- Generic and specific types embeddings generation:
  - We used the ontology hierarchy of Wikidata/DBpedia KG.
  - The hierarchies were flattened into a sequence of ontology terms that were used as input for embedding learning.
  - We used the Word2Vec approach [1] in which the ”CBOW” neural network with a layer size of 100 was used for generating type embedding.

[1] Mikolov, T et.al, Efficient estimation of word representations in vector space, 2013


Type Hierarchy
A
B
F
C
D
Question Text
'What position did William Carmichael replace John Jay in?'
Example Generation
Generic Type
Specific Type
Z+
A
B
B
C
Z-
A
F
B
D
Train Classifier
Question Text
0.6
0.1
-0.7
0.5
Generic Type
0.9
0.6
-0.6
0.3
Specific Type
0.3
0.2
-0.4
0.3
Concatenate
Average
0.6
0.1
-0.7
0.5
0.6
0.4
-0.5
0.3
Classifier

Methodology

- Question specific type prediction:
  - We modeled this problem as a binary classification problem.
  - The given generic type and specific type that match the question are the positive examples.
  - We generated negative examples by replacing the specific type in the positive instances by another type that shares the same parent type in the hierarchy but not a correct specific type.

Methodology

- Question specific type prediction:
  - The positive instances were labeled as “1” (output variable) and the negative instances were labeled as “0”.
  - The random forest model was trained on the training set and achieved an accuracy of 0.89.

Conclusion

- Most state-of-the-art classifiers have limited capabilities in granular classification tasks.
- We propose a framework that focuses on hierarchical type prediction for engineering features that can subsequently improve upon the already existing methodologies such as Random Forests.
- We report performance of the framework on granular categorization data for ‘resource’ categories using ontologies classes WikiData/DBpedia using NDCG@5 and NDCG@10 metrics.
- For validation, we achieve 0.621 and 0.606, respectively.