Improving Search for ExploreCourses

Sam Redmond, Eddie Wang

Overview

The existing implementation of course search for ExploreCourses relies on naïve substring matching. We build an improved search model for the Stanford course dataset by composing document vector and word vector models.

Objective:	Improve relevance of course search results

Dataset and Processing

2015-2016 course listings from ExploreCourses
- 14K courses (code, title, description)
- 37K unique words
- ~1.3M tokens

We preprocess each course
- Split sentences with NLTK
- Parse tokens with Penn-Treebank tokenization
- Compute and condense the most likely bigrams

Each sentence is labelled with the course code and a unique identifier for training.
For example:

```
Title: machine learning
Description: topics: statistical pattern
recognition, linear ...
```

becomes

```
([['machine_learning'], ['topics', ':',
'statistical', 'pattern_recognition',
',', 'linear', ...], ...], ["cs229"])
```

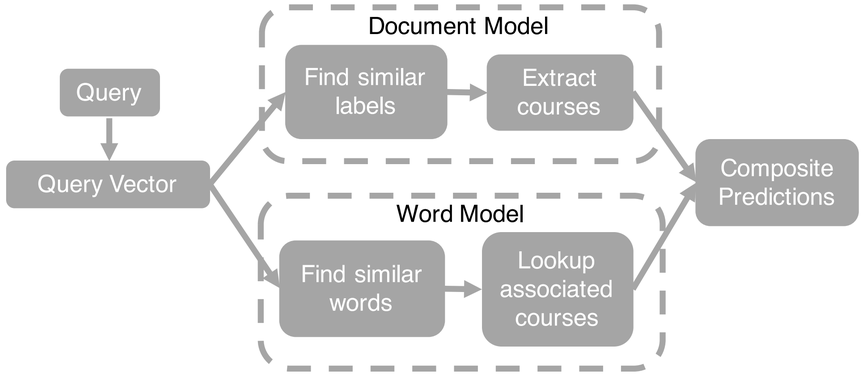
Methods

Training

Document model
- Obtain document and word vectors by training a distributed memory (PV-DM) paragraph2vec model
- Course descriptions are mapped to dense, fixed-length vectors
- Word vectors initialized to pre-trained values (randomly if unseen)
- Pre-trained tokens learn at 1/5 ordinary learning rate
- Repeatedly train until the word and label vectors reach convergence.

Word model
- Each word maps to its most similar courses
- For a given word, each course is scored by: $\frac{f_{course}}{tokens_{course}} \div \frac{f_{total}}{tokens_{total}}$, similar to the relative frequency metric used in Naïve Bayes text

Suggesting



Querying
- Convert query string to query vector using trained paragraph2vec model
- Document model: find closest label(s) by cosine similarity, then extract course codes from label(s)
- Word model: find similar words to the query vector, then look up associated courses
- Average results together, weighted by confidence, to produce final result

Evaluation and Results

Qualitative
- On average, our composite model outperforms ExploreCourses.
- Worst case: not worse than ExploreCourses

Sample queries for which the difference is especially apparent

Query	ExploreCourses	Composite Model
Computer Programming	Biophysics of Multi-cellular Systems Physics-Based Simulation Computational Genomics	Introduction to Computing Principles Programming Service Project Introduction to Computers
Violin Lessons	None	Introductory Violin Class Violin Advanced Violin
Linear Algebra	Advanced Feedback Control Design Numerical Methods in Engineering Mechanical Vibrations	Calculus Modern Algebra Linear Algebra and Differential Calculus

Top three search results for the ExploreCourses model and our composite model run against three sample queries.

Quantitative
- Asked 20 students to create ten queries within several categories and score the results out of 5.
- Other user interaction metrics are withheld for privacy reasons

Category	ExploreCourses	Paragraph2Vec	Word2Vec	Composite
Find a specific course	3.4	2.5	4.2	4.4
Find courses on a specific topic	2.2	3.1	3.8	4.0
Find courses on a general topic	1.3	3.6	3.2	3.9
Total	**2.3**	**3.1**	**3.7**	**4.1**

Average scores (out of 5) given by n=20 students assessing performance of different models across a variety of search categories