University of Windsor

School of Computer Science
Faculty of Science

COMP-8730: Natural Language Processing & Understanding
Winter 2021

#	Title	Due Date	Grade Release Date
4	**Assignment 02**	March. 01, 2021, AoE	March. 08, 2021, AoE

The objectives of the assignments are to practice on topics covered in the lectures as well as improving the student's critical thinking and problem-solving skills in ad hoc topics that are closely related but not covered in the lectures. Lecture assignments also help students with research skills, including the ability to access, retrieve, and evaluate information (information literacy).

Lecture Assignment

We explained two approach to semantics, namely i) *lexical* semantics based on lexical similarity and determined manually by linguistics, and ii) *vector* semantics that is based on the idea of distributional semantics, i.e., semantic similarities between linguistic items based on their distributional properties in large samples of language data. In this assignment, we want to evaluate how these two are correlated. For simplicity, we argue that lexical semantics are the golden standard about the semantics of the words based on which we evaluate the results of vector semantics. In another word, the more the results of vector semantics are close to the lexical semantics, the better.

Given a golden standard *G* and a *large* corpus of text *C* for English language, calculate the average Information Retrieval (IR) metric *m* of top-k similar words retrieved by the vector semantics based on method *v.*

a) *G*: Report the evaluation results based on 2 golden standards WordSim-353[1] and SimLex-999[2].
b) *C*: Report the evaluation results based on 2 large corpus *from different genres* available in NLTK or Gensim libraries.
c) *v*: Report the evaluation results of methods Term-Document, Term-Term, TF-iDF, Word2Vec using the *cosine* similarity. These methods are also called baselines.
d) *top-k*: Report the evaluation results for top-10, i.e., k=10.
e) *m*: Report the evaluation results based on average MAP and nDCG using PyTrec_Eval[3].

Evaluation Methodology

1) We select SimLex-999 as our golden truth.
    a. For each word *w*, we order the top-10 similar words to *w* as golden list for *w*. Note that we may have list of different sizes for each word *w*. For instance, for 'soccer' we may hape 3 most similar words and for 'apple' we may have 20 most similar words.
    b. When the size is smaller than 10, we try to expand it by transitivity rule, i.e., *w* similar-to *a*, *a* similar-to *b*, then *w* similar-to *b*. If we don't reach to top-10, we leave it as it is.
    c. When the size is greater than 10, we truncate the list to top-10.
    d. Let's call the golden top-10 similar words to *w* as top-k-G[*w*]; *k*=10.

[1] Agirre, Eneko, et al. "A study on similarity and relatedness using distributional and wordnet-based approaches." (2009).
[2] Hill, Felix, Roi Reichart, and Anna Korhonen. "Simlex-999: Evaluating semantic models with (genuine) similarity estimation." Computational Linguistics 41.4 (2015): 665-695.
[3] Van Gysel, Christophe, and Maarten de Rijke. "Pytrec_eval: An extremely fast python interface to trec_eval." The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018.

   e. Note the the top-10 list is ordered descending based on the similarity scores in SimLex-999 which are between 0 and 10.

2) We pick 'news' genre of the 'Brown' corpus in NLTK library as our large corpus (it's not large compared to the nowadays corpora. But for the sake of this assignment, it is acceptable).

3) We pick Word2Vec as our vector semantic method (baseline).
   a. We train Word2Vec on our large corpus, 'news' collection of Brown corpus.
   b. We report the running parameters of Word2Vec.
      i. Context window size = 5
      ii. Vector size = 100
      iii. Iteration number = 100

4) For each word *w* in our golden standard SimLex-999, we find the top-10 most similar words according to *cosine* similarities of Word2Vec vectors.
   a. If *w* is not in our large corpus, then it is unseen words and an instance of OOV. In this assignment, we simply ignore this word.
   b. If *w* is in our large corpus, then there are top-10 most similar words that are ordered based on descending order of cosine similarity scores.
   c. Let's call the top-10 most similar words of *w* based on Word2Vec as top-k-w2v[*w*]; *k*=10.

5) Now we have to compare top-k-G[*w*] and top-k-w2v[*w*] for all w that exists both in golden standard and our large corpus.
   a. We give these two list to PyTrec_Eval. This library is to calculate the IR metrics assuming there are two lists of results for a query, one list as the true list and one list as the prediction list. In our assignment, the query is the word *w* and the true list is top-k-G[*w*] and the prediction list is top-k-w2v[*w*].
   b. We ask PyTrec_Eval to calculate 'MAP' and 'nDCG' as our metrics *m*. The result is for each for *w*.
   c. We calculate the average of 'MAP' and 'nDCG' on all words.
   d. We report the results on a bar chart.

6) We have to repeat the procedure 3 to 5 for other vector semantic methods (baselines).
7) We have to repeat the procedure 1 to 5 for the second golden standard WordSim-353.

In total, we have
G:{SimLex-999, WordSim-353} × C:{'news', 'romance'} × v: {Term-Document, Term-Term, TF-iDF, Word2Vec} × m:{MAP, nDCG} = 32 bars!

Findings

In the end, we have to analyze the results to answer our original research questions (RQs):

- *RQ1: do vector semantic methods capture the lexical semantics among the words?*
- *RQ2: Which baseline is more effective (higher performance metric)?*
- *RQ3: (optional) Which baseline is more efficient (faster)?*

Submission Guidelines

- Submission must be written as a report in English, in the current ACM two-column conference format in LaTeX. Overleaf templates are available from the ACM Website (use the "sigconf" proceedings template).

- The report must be 2 pages in length, no more no less, including figures, tables, references, and single authored by the student.

- The implementations (code) should be available in an online repo (preferably Github) and the link should be mentioned as a footnote to the report's title. See the example below. The results reported in the report must be reproducible (multiple runs same result).

- Submission must be in one single zip file named COMP8730_Assign02_UWindId.zip, including:
  1. the LaTeX files
  2. the pdf file

A sample submission has been attached to this manual in Blackboard, also available online here.