University of Windsor
Faculty of Science
School of Computer Science

COMP-8730
Natural Language Processing and Understanding
Winter 2024

#	Title	Due Date	Grade Release Date
Assignment 02	Vector vs. Lexical Semantics	April 31, AoE	April 8

The objectives of the assignments are to practice on topics covered in the lectures as well as improve 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, *which is not necessarily true in practice*.

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 the golden standards SimLex-999[1].
b) *C*: Report the evaluation results based on 2 large corpus from *different* genres available in NLTK[2] libraries.
c) *v*: Report the evaluation results of methods TF-iDF[3], Word2Vec[4] 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 nDCG[5] using pytrec-eval-terrier[6].

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 have 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.
   e. Note the the top-10 list is ordered descending based on the similarity scores in G.

2) We pick *C* as our large corpus.

---

[1] Hill, Felix, Roi Reichart, and Anna Korhonen. "Simlex-999: Evaluating semantic models with (genuine) similarity estimation." Computational Linguistics 41.4 (2015): 665-695.
[2] https://www.nltk.org/
[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[4] https://radimrehurek.com/gensim/models/word2vec.html
[5] https://en.wikipedia.org/wiki/Discounted_cumulative_gain
[6] 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.

3) We pick *v* method (baseline).
   a. We train *v* on *C*.
      i. We report the running parameters of *v* if any.
      ii. For Word2Vec, we run for context window size {1, 2, 5, 10}, vector size {10, 50, 100, 300}, and iteration number = 1000.

4) For each word *w* in our golden standard G, we find the top-10 most similar words according to *cosine* similarities of vectors based on method *v*.
   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 *v* as top-k-*v*[*w*]; *k*=10.

5) Now we have to compare top-k-G[*w*] and top-k-*v*[*w*] for all *w* that exists both in golden standard and our large corpus.
   a. We ask pytrec_eval to calculate 'nDCG' as our metric *m*. The result is for each for *w*.
   b. We calculate the average of 'nDCG' on all words.
   c. We report the results on a bar chart.

6) We have to repeat the procedure 3 to 5 for all methods *v* (baselines).

For Word2Vec, we have 4×4 different running settings. We pick the best setting according to the highest nDCG. Finally, we have
G:{SimLex-999} × C:{'news', 'romance'} × v: {TF-iDF, $\text{Word2Vec}_{\text{best}}$} × m:{nDCG} = 4 bars!

You have to come up with the best way to represent and explain the findings.

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 [7] are available from the ACM Website [8] (use the sigconf proceedings template).
- The report must be 1 page in length, no more no less, including figures, tables, references, and single-authored by the student.
- The 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 with the same setting should result in the same results.)
- Submission must be in one single zip file named `COMP8730_Assign03_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[9].

[7] https://www.overleaf.com/gallery/tagged/acm-official
[8] https://www.acm.org/publications/proceedings-template
[9] https://www.overleaf.com/read/nwfkjczgvxzn