CSC 485H/2501H, Fall 2021: Assignment 2

Family name: Yang

Given name: Jamin

Student #: 1005186558

Date: 11/5/2021

I declare that this assignment, both my paper and electronic submissions, is my own work, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters and the Code of Student Conduct.

Signature: Jamin Yang

Part 1:

1 d) For word-sense disambiguation and more importantly, the LESK algorithm, having more data of relevant word usage in the signature set helps improve accuracy. This is because with a larger signature set (with meaningful examples and definitions), there is a higher chance for overlap between the context and signature sets. This allows LESK_ext to predict the correct sense with greater accuracy than LESK in the general case.

1 g)

lesk_cos_onesided performed better than lesk_cos by more than 5% (refer to accuracies below). This difference in accuracy is because the vectors in lesk_cos include both context and signature words while lesk_cos_onesided only includes the context words.

For example, say we wanted to predict the correct sense for "art" in the context sentence "The art of change-ringing is peculiar to the English language, and like most English peculiarities, unintelligible to the rest of the world." Well, lesk_cos would have wordforms like "computer", "printmaking", "parts", and "software" in their vectors (due to signature) which are unrelated to our context sentence, while lesk_cos_onesided would only have wordforms related to the context sentence like "English", "change-ringing," "peculiar". As a result, lesk_cos_onesided vectors are more relevant and similar to the context sentence while lesk_cos vectors have some differences resulting from the inclusion of senses with non-relevant contexts in the signature set's definitions and explanations.

Since lesk_cos_onesided vectors are more relevant to the context sentence, the function predicts the correct sense better on average compared to lesk_cos.

1 h)

Cosine similarity is around 0.408 which tells us the sets have weak similarity.

Set intersection = {buffalo} with cardinality = 1 (weak similarity since only one word out of 6 is in the intersection).

Therefore, set intersection is similar to cosine similarity in the way that they can both measure the similarity of two sets/vectors.

1 i)

mfs: 41.6%

lesk: 39.3%

lesk_ext: 45.9%

lesk_cos: 38.3%

lesk_cos_onesided: 44.0%

lesk_w2v: 47.9%

1 j)

Before-After comparison:

mfs: 41.6% -> 41.6%

lesk: 39.3% -> 37.8%

lesk_ext: 45.9% -> 45.9%

lesk_cos: 38.3% -> 36.0%

lesk_cos_onesided: 44.0% -> 43.8%

lesk_w2v: 47.9% -> 47.4%

Observations:

lesk and lesk_cos accuracies dropped by 1-2%. lesk_cos_onesided and lesk_w2v dropped by less than half a percent. Other methods did not drop.
mfs is unaffected by upper-case/lower-case (solely uses lemmas) so accuracy doesn't change.
lesk_w2v not affected as much because it averages all relevant vectors.

Explanation:

Sometimes, words with upper-case letters have different meanings compared to their lower case versions. For instance, proper nouns are more likely to have an upper-case letter than a common noun. In this regard, grouping words with upper-case letters with their lower-case versions can result in misclassification (thus, lower accuracy).

Example:

For example, "Association" in "the Association of Physics and Chemistry…" is likely to be about a formal organization of people or groups of people. On the other hand, "association" in "the association of physics and chemistry…" is likely to refer to the relationship between physics and chemistry.

Part 2:

a) Yes, context matters. Let's say we were trying to predict the correct sense for the word "thing". Well, this word is incredibly ambiguous because it can be used in many contexts. Wordnet suggests that "thing" can mean an action, an artifact, an event, a special objective or situation, etc., for a total of 12 senses. Our word order-invariant methods in Q1 would never be able to disambiguate "thing" based on wordform and lemma alone because word2vec would return the same vector for "thing" regardless of how it is placed in the sentence. If the word was used in an uncommon context (example: "watch out for that thing" could mean a persistent illogical feeling of desire if someone was pointing at something very attractive to them), it would be essentially impossible for the method to pinpoint that particular sense. This problem of word sense disambiguation can be attenuated using context, as analyzing the words preceding (and after) could help tell you what particular sense it is.

Of course, "thing" isn't the only word where context matters. In the general case, any ambiguous word with very different contexts like bark (bark of a tree, a dog's barking) or play (to play, or a theatre play) will make the Q1 methods have a hard time correctly predicting their sense.

c) When the run_bert function is called to tokenize the batch for BERT, it pads the tokenized sentences up to the longest batch sentence. Hence, mixing short and long sentences increases run time due to the relative increased padding. In comparison, there is minimal padding when sentences of similar length of batched.

e)

Our code model was trained on certain words in certain contexts, e.g., our corpus in Q2 gather_sense_vectors. If we were to throw an arbitrary sentence with words not in the dictionary returned by gather_sense_vectors, it is likely to predict the wrong sense (Since there are no sense vectors, Q2 uses MFS which only assumes the most frequent sense. Bad if our sense is used less frequently).

Of course, depending on the context of the arbitrary sentence, we would still have problems even if it was in the dictionary returned by gather_sense_vectors. This is because it was trained on a certain corpus, so the context might be entirely different, resulting in a misclassification once again (In other words, the predicted senses will be similar to the ones in the corpus. Likely to result in a misclassification if sentence uses the word in contexts very different than the ones in the corpus).

Q2:

Accuracy: 47.9%-48%