Assignment 1: Distributional Similarity

Submitters: Ori Braverman 318917010

Yaniv Zimmer 318849908

Generating lists of the most similar words

The code is in “section1.py”, the results in “generating_lists_of_the_most_similar_words.txt”.

Polysemous Words

The code is in “section1.py”, the results in “polysemous_words.txt”

1) Which three polysemous words belong in the first group, and what are their neighbors?

Answer - Polysemous words that the top-10 neighbors of each word reflect both word meanings:

- mouse (both computer mouse neighbors and the animal neighbors)

- dish (bot meaning – as a kind of plate or a meal)

- bat (bot meaning – as a baseball bat and a flying rodent)

2) Which three polysemous words belong in the second group, what are the possible

senses for each word, and which of the senses was reflected in the top-10 neighbors?

Answer - Polysemous words that the top-10 neighbors of each word reflect both word meanings:

- apple (only neighbors correspond to food)

- pen (only neighbors correspond to a writing implement)

- frame (only neighbors correspond to a surrounding for a picture)

3) Can you explain why the second group words neighbors reflect only one sense?
Answer – the reason that the second group words neighbors reflect only one sense could be:

- The learning data – the training data that the model learned from was more biased towards one side of the meaning. In other words, the frequency of usage of words in one of the meanings was more dominant.

- Common usage – it can be that the other meaning is not frequently used in the language.

Synonyms and Antonyms

The code is in "section1.py", the results in "synonyms_and_antonyms.txt"

1. Write your triplet.

   w1, w2, w3 = 'Happy', 'Gay', 'Sad'

   sim w1, w2: 0.18161133

   sim w1, w3: 0.43675756

   w1, w2, w3 = 'rose', 'soar', 'decreased'

   sim w1, w2: 0.45347553

   sim w1, w3: 0.5525813

2. Can you explain this behavior in which the antonyms are more similar than the synonyms?

   The word usage of w2 is low in the corpus: soar is a high word in the English language and gay is a polysemous word that the meaning correlated to happy hardly appears. Therefore, w1 and w3 can occur more frequently.

The Effect of Different Corpora

The code is in “section1.py”, the results in “effect_of_different_corpora.txt”

1. Which words did you find?

   car, dog, apple, rain, weekend, report, pillow, outlet, manufacture, truck

2. For the second case, describe in words the difference in neighbors you observe.

   The word: course The difference in neighbors: the top 10 neighbors based on the wiki corpus were words like "courses," "golf," "going," and "turn. The context is more about education, sports and directions, however, in twitter’s list the words were "sure," "yeah," "well," and "yep" these words are slang, casual conversation. More words: thanks, king, smiley, haha, love The difference in neighbors: for these words a similar pattern can be observed. In the Wiki corpus, the top 10 neighbors often reflect a more formal or diverse context, while in the Twitter corpus, the neighbors tend to be more casual, slang and directly related to online communication. This difference between the Wiki and Twitter corpus emphasizes the change between formal language to the ongoing, changing language of teenagers and shortcuts (like thanx, thx, btw in thanks example).

For example, when we picked the word “haha”:

```
Using wiki model:
The most similar words to 'haha' are:
speakerphones: 0.48436424136161804
kī: 0.466961145401001
kuni: 0.45945024490356445
chichi: 0.4517125189304352
jutro: 0.4502652585506439
custalow: 0.44933298230171204
cyanea: 0.4402952790260315
chishima: 0.4356171488761902
insh: 0.43075433373451233
iesu: 0.4305413067340851
```

```
Using twitter model:
The most similar words to 'haha' are:
hahaha: 0.9729723334312439
hahah: 0.9502511620521545
hahahaha: 0.9211272597312927
hahahah: 0.9033352732658386
hahha: 0.8968350291252136
hahaa: 0.8962197303771973
hahahha: 0.878961443901062
hehe: 0.8726717233657837
ahaha: 0.8685161471366882
<lolface>: 0.8659974932670593
```

The Wiki model didn’t recognize it on the corpus while in twitter there were a lot of variations and even an encoding of emoji (lolface = smily face).

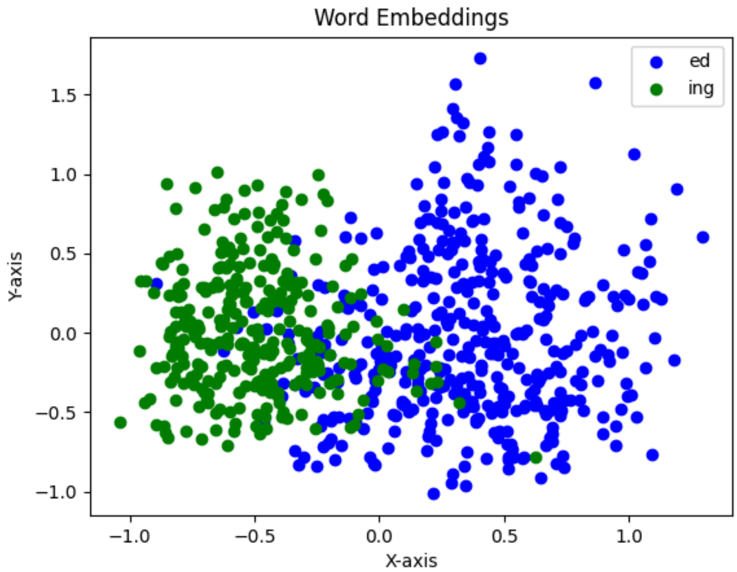
3) What was your strategy (either automatic or manual) for finding these 10 words?

At first, sampled random words from : Random Word Generator
We also tried to consider words with double meaning\ context dependent or very popular in youth internet culture (as haha) because we expect them to act differently in wiki and twitter contexBut most of the words were similar both on wiki and twitter

The conclusion that we came to was that slang from twitter acts very different in wiki,
This slang will appear rarely in eloquent language on wiki so in that way I will achieve different neighbors on each model.

Plotting words in 2D

The code is in "plot_embedding_filer_vocab_.py", the results in "asset/ed_vs_ing.txt"



Are the colors separated? why or why not? is the information included in the model?

Answer:

The colors are not fully separated. The overlapping between the words can occur due to semantic similarity between the words, even if they have different suffix.

For example, after looking at the 708 words vocabulary, we found similar words with different suffixes, so it may be correct if they have similar vectors.

The words were: played – playing, ended – ending, need – needed, added – adding.

So to conclude, the reason that the color may not be separated is due to morphology, different forms of the word.

Word-similarities in Large Language Model

Related words

Lists of words provided by ChatGPT:

Car: ["vehicle", "automobile", "truck", "sedan", "motorcar", "ride", "transportation", "drive", "roadster", "convertible", "wheels", "automotive", "engine", "speed", "traffic", "commute", "journey", "gas", "highway", "transport"]

Dog: ["canine", "puppy", "pet", "hound", "pooch", "animal", "companion", "bark", "wag", "leash", "fur", "loyal", "breed", "tail", "fetch", "paws", "snout", "collar", "domesticated", "muzzle"]

Apple: ["fruit", "orchard", "tree", "crisp", "sweet", "red", "green", "orchard", "healthy", "snack", "juice", "core", "cider", "pie", "harvest", "Granny Smith", "Gala", "pomaceous", "nutrition", "delicious"]

Face: ["facial", "features", "eyes", "nose", "mouth", "expression", "smile", "lips", "forehead", "chin", "cheeks", "beauty", "skin", "appearance", "identity", "portrait", "gaze", "look", "emotion", "complexion"]

Vice: ["wrongdoing", "immorality", "sin", "corruption", "evil", "misconduct", "decadence", "depravity", "wickedness", "transgression", "offense", "flaw", "fault", "vice versa", "bad habit", "immorality", "sinfulness", "iniquity", "weakness", "wrongdoing"]

Examine qualitatively the outputs of the two methods. How do they compare in terms of accuracy? in terms of diversity? are they all words, or are some multi-word phrases? if there are phrases, can you make it produce only words? In addition, try identifying different types of similarity that are reflected in the lists, give some examples for each type, and examine whether these types differ between the word2vec output and the ChatGPT output.

Answer:

Accuracy: Both ChatGPT and word2vec provided accurate results, but that may have been different. Word2vec has pre-trained vector embeddings and may struggle to generalize words that do not appear in his dictionary to the correct context (because of his limited size). On the other hand, ChatGPT can easily manage unknown words because it is substantially bigger and trained on more data.

Diversity: ChatGPT managed to provide more diverse results.

For example, in the car case, word2vec provided mostly car names and little about types of cars and ChatGPT mentioned car types, car features, car parts, actions, where to ride it.

Therefore, ChatGPT managed to analyze in a more detailed way the words associated, and word2vec in a more shallow way that considers the word location in English sentences and not necessarily its true meaning.

Word vs. multi-word phrases: ChatGPT output has only individual words, but word2vec doesn't.

For example, in 'face' similarities: 'Rutter_wrenched', 'Brownell_inherits', 'WabiSabiLabi_auctions', 'gauzy_handkerchiefs'. Or our favorite: 'Mother_Nature_Gino_Izzi'.

Types of Similarity:

Brend association:

Word2vec: 'car' is associated with 'SUV', 'minivan', 'truck'.

ChatGPT: 'car' is associated with "truck", "sedan", "motorcar".

Fruit association:

Word2vec: 'apple' is associated with 'pear', 'fruit', 'berry', 'pears', 'strawberry', 'peach'.

ChatGPT: 'apple' is associated with "fruit", "Granny Smith", "pomaceous".

Now select two of the words and increase the number of neighbors from 20 to 100, for both

The code is in “section1.py”, the results in “related_words.txt”

ChatGPT and word2vec, what trends do you observe?
* The output is in file related_words.txt

Answer:

In the dog word context, the increase of number didn’t affect the word2vec list – almost all the words are dog breeds. What changed the most was that the ChatGPT list gained substantial amounts of dog breeds (collie, shepherd, terrier, spaniel, bulldog, retriever, poodle, dachshund), dog actions (howl, growl, fetch, play, loyal companion, barking, obedient) and contains more diversity and more similarity to word2vec.

In the apple word context, ChatGPT managed to give more fruits like apple in contrast to the 20-word list (orange, lemon, lime, grape, banana, pineapple, mango, kiwi). The word2vec remained superficial, it gave only fruits in his list. In addition, ChatGPT expanded the apple uses – how to get it, what products you can make with it, how you can make apples (harvest, autumn, cider, pie, jam, sauce, slice, peel) (some of the previous uses from the 20-words list remained).

Note: the first prompt (the instruction you give ChatGPT) you try may or may not work well, you are encouraged to experiment with different prompts. In your report, write examples for a few prompts that you tried and which prompt worked best for you.

Answer:

<u>This is the first prompt I tried:</u> please generate in vector form [,,,] 100 most similar words to dog and apple.
<u>The problem:</u> he didn’t give 100 words each (only 25) and the output wasn’t in vector form.

<u>Other prompt:</u> please generate 100 most similar words to dog and apple. The output you should provide must be in that form: [word1, word2, word3, etc]

<u>The problem:</u> It didn't give 100 words for each word.

<u>Final prompt:</u> please generate 100 most similar words for each of the words: dog and apple. The output you should provide must be in that form: list: word_chosen: [word1, word2, word3, etc] word_chosen: [word1, word2, word3, etc] word_chosen: [word1, word2, word3, etc].

Synonyms and Antonyms

Can you get ChatGPT to reliably produce synonyms? Which prompt did you use?

Can you get ChatGPT to reliably produce antonyms? Which prompt did you use?

Answer:

As well known, ChatGPT can't provide reliable answers to any kind of question, as it does not guarantee reliable answers, as well as synonyms and antonyms. The answer generation depends on various factors like the complexity of the task, or how specific it is.

The prompt: please provide me in reliable way synonyms and antonyms (5 each). thank you.

ChatGPT answer: Certainly! Here are five synonyms and five antonyms for the word "happy":

Synonyms for "Happy":

1. Joyful
2. Content
3. Pleased
4. Delighted
5. Ecstatic

Antonyms for "Happy":

1. Sad
2. Unhappy
3. Miserable
4. Depressed
5. Sorrowful

<u>Polysemy</u>

How does ChatGPT behave when you ask it for similarities for the polysemous words you found in the word2vec part? In what senses is it similar, and in what senses is it different from word2vec?

Answer:

<u>The prompt:</u> please provide me 10 words for each word in the list that are similar to this word.

list = ["apple", "mouse", "pen", "dish", "frame", "cap", "drive", "satellite", "bat", "bank", "rose"].

please provide the output in a form of:

word_name: [w1, w2, w3]

word_name: [w1, w2, w3]

…

thank you.

<u>ChatGPT answer:</u>

Certainly! Here are 10 words similar to each word in your list:

1. apple: [fruit, orchard, pear, peach, berry, citrus, pomegranate, cherry, plum, apricot]

apple (the fruit) and apple (the company)

2. mouse: [rodent, computer, device, rat, click, pointer, cursor, trackpad, scroll, wireless]

mouse (the animal) and mouse (the device)

3. pen: [writing, ink, pencil, paper, ballpoint, notebook, quill, marker, sketch, stationary]

pen (a writing implement) and pen (the action of writing)

4. dish: [plate, cuisine, bowl, food, meal, serving, platter, cookware, recipe, dinnerware]

dish (a kind of plate) and dish (a meal)

5. frame: [picture, border, structure, framework, window, outline, boundary, edge, framing, composition]

frame (a surrounding for a picture) and frame (falsely incriminate)

6. cap: [hat, lid, cover, headwear, crown, beanie, baseball, snapback, visor, beret]

cap (the top of a pen or bottle) and cap (a type of headgear)

7. drive: [motivation, trip, journey, urge, ambition, push, ride, excursion, initiative, momentum]

drive (control a vehicle) and drive (a short private roadway)

8. satellite: [orbit, space, communication, telescope, broadcast, navigation, spacecraft, celestial, signal, link]

satellite (an orbiting space craft) and satellite (a country or state dependent on a more powerful one)

9. bat: [racket, mammal, hit, swing, flying, cricket, baseball, paddle, flittermouse, club]

bat (the animal) and bat (the sports equipment)

10. bank: [financial, institution, river, shore, slope, finance, account, money, lender, deposit]

bank (a commercial enterprise concerned with money) and bank (the side of a river)

11. rose: [flower, blossom, petal, bouquet, garden, red, love, romance, bloom, fragrance]

rose (the past form of rise) and rose (a flower)

I hope this helps!

ChatGPT missed some of the polysemy words like: apple, frame and rose, but he did a lot better in finding polysemous words than word2vec.

Mean Average Precision (MAP) evaluation

First, lets see the 4 lists from before:

Word2vec:

Dog: ['dogs', 'puppy', 'pit_bull', 'pooch', 'cat', 'golden_retriever', 'German_shepherd', 'Rottweiler', 'beagle', 'pup', 'Pomeranian', 'canines', 'chihuahua', 'pet', 'schnauzer', 'Shih_Tzu', 'pit_bull_mix', 'dachshund', 'Labrador_retriever', 'border_collie']

Apple: ['apples', 'pear', 'fruit', 'berry', 'pears', 'strawberry', 'peach', 'potato', 'grape', 'blueberry', 'cherries', 'mango', 'apricot', 'melon', 'almond', 'Granny_Smiths', 'grapes', 'peaches', 'pumpkin', 'apricots']

ChatGPT:
Dog: ["canine", "puppy", "pet", "hound", "pooch", "animal", "companion", "bark", "wag", "leash", "fur", "loyal", "breed", "tail", "fetch", "paws", "snout", "collar", "domesticated", "muzzle"]

Apple: ["fruit", "orchard", "tree", "crisp", "sweet", "red", "green", "orchard", "healthy", "snack", "juice", "core", "cider", "pie", "harvest", "Granny Smith", "Gala", "pomaceous", "nutrition", "delicious"]

Note: we will color each judgment with a different color.

Specifically, for each candidate word in the similarity lists, make two judgments:

1. Whether you judge the candidate word to be **topically** related to the target word.

2. Whether you judge the candidate word to be in the same **semantic** class as the target word.

-this color if both apply

The semantic class that we chose for the word 'dog' is animals and the semantic class for the word 'apple' is fruits. We noticed that the semantic class is also topic related to each of the words so this could also be interpreted as a topic related sentence.

For apple in word2vec all similar words are fruits, and one is a type of apple (Granny_Smiths) which we also decided to consider as a fruit

We observed that ChatGpt tends to choose words that are topically related, but not necessarily semantic related

Dogs:

$maP = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{Ni}\frac{1_{ik}}{Ni}$ where 1_{ik} is 1 if the word w_{ik} in the list of proposed similar words relate to the word w_i in the input words.

Word2Vec

1. Topically-
    a. dogs = 20/20 = 1
    b. Apples = 20/20=1

    ⇒MaP = (1+1)/2 = 1
2. Semantic Class
    a. dogs = 20/20 = 1
    b. Apples = 20/20=1

    ⇒MaP = (1+1)/2 = 1

ChatGPT

3. Topically-
    a. dogs = 16/20 = 0.8
    b. Apples = 20/20=1

    ⇒MaP = (0.8+1)/2 = 0.9
4. Semantic Class
    a. dogs = 6/20 = 0.3
    b. Apples = 4/20=0.2

    ⇒MaP = (0.3+0.2)/2 = 0.25