Tamir Moshiashvili

Amos Maimon

Report:

1. Statistics:  
   a. threshold:  
    - filtered the words that appeared below 100 times (those words has no vector)  
   b. number of words that occurred at least 100 times - 14490  
   c. number of features for each word – at most, 50
2. 20-most similar words for each target-word is in a separate file, called "most\_common\_words.csv".  
     
   \* Conclusions for co-occurrence-1:  
    similar words are at most semantic-related because when the features are the  
    sentence's words, there might be a lot of features that can be used like the word,  
    thus the similar words has the same semantics.  
    But, also the features will contain topic-related words, so sometimes there will be  
    words that connected by topic.  
     
    For example, for car, most of the similar words are types of cars or a vehicle,  
    and for piano, there are few words that connected topically.  
     
   \* Conclusions for co-occurrence-2:  
    similar words may be both semantic and topic related because the features taken  
    from the window, so topic-related words are in the window (because we  
    ignore functional-words), but also semantic-related words (words that are close  
    to the target word in the sentence).  
     
    For example, in both car and piano, the amount of topic-related and  
    semantic-related is almost the same.  
     
   \* Conclusions for co-occurrence-3:  
    similar words are at most topic-related words, because the features are taken  
    from the dependency-tree, there is more chance that the features will be  
    connected by that topically.  
    But there might happen that a word will contain as feature a word that has the  
    semantics.  
     
    For example, in car, most of the similar words are topic-related, and there are  
    few words that are semantic-related,  
    and for piano, there is same amount for both.
3. 20-most common features for each target word in each co-occurrence is in a separate file, named "most\_common\_first\_order.csv"  
   1. first co- occurrence:

In the next item we found for the word "car", as example, in the first co-occurrence type some drivers name, and also some car manufacturers names. It is match the fact that in the common features to this word we find a lot of personal pronoun, and the words "design" & "sport", this, in our opinion, cause a match for the words above.

Also, for the word "piano" – we found a lot of features are also found in the similar words list. We assume that it occurs because this co- occurrence take the all sentence.

We found this phenomenal in others targets words

* 1. Second co- occurrence:

In the next item we found for the word "car", as example, for this co-occurrence type. That there is some words that not similar in topic or in semantic ways, look on its features reveals that it is occur because some the common feature have multi meaning (like "park") or to generic (like, "sell").

Also, we found in other target word a phenomenal – some similarity has been detected between words that used in the same sentences with a jump of word (or two). Example: some direction names ("north-south") detected as similar to the word "bus". It makes sense that a feature like "connect" and "route" are connected this words to "bus" because there common to follow them in the jump with this features words.

* 1. Third co – occurrence:

We found a phenomenal that targets words in this co -occurrence type that have a verbs that usually use with them can drag the similarity to adjunctives that use with them. Example: for "car" there is a lot of negative words in the similarity list- "deadly", "fatal", "freak", and so on, we assume that it occur because the features with the words "hit", "accident", "bomb", "crash" , are so common.

* 1. For all:

we found that if the target word had multi meaning, the features can drag the common meaning in the corpus as the only one. Example: the word "fox" in all the co-occurrence types was construed as the news channel, and not as the animal. We can see in the features list – that the most common feature are agreed with that distinction.

1. For car:

Topic-related marked in **red**

Semantic-related marked in **blue**

|  |  |  |  |
| --- | --- | --- | --- |
| car | Co-occur1 | Co-occur2 | Co-occur3 |
|  | andretti | auto | airplane |
|  | bmw | automobile | amusement |
|  | chevrolet | champ | auto |
|  | fia | constructor | automobile |
|  | ford | driver | challenger |
|  | gt | endurance | deadly |
|  | harvick | fia | drunk |
|  | midget | formula | fatal |
|  | motorcycle | indy | freak |
|  | motorsport | lap | hunting |
|  | motorsports | motor | indoor |
|  | nascar | motorcycle | motorcycle |
|  | nissan | nascar | plane |
|  | suv | off-road | racing |
|  | tdi | racing | rally |
|  | toyota | stock | skiing |
|  | traction | thoroughbred | traffic |
|  | truck | toyota | tragic |
|  | vauxhall | truck | truck |
|  | waltrip | yacht | turbo |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Co-occurrence-1 | | Co-occurrence-2 | | Co-occurrence-3 | |
| topic | semantic | Topic | Semantic | Topic | semantic |
| 2/20 | 18/20 | 7/20 | 12/20 | 13/20 | 7/20 |
| 10% | 90% | 35% | 60% | 65% | 35% |

Insights:

Most semantic-related words are type of a car or transportation-object.

Topic-related words are words that connected to car in different ways, like traffic, lap, drunk and so on.

For piano:

Topic-related marked in **red**

Semantic-related marked in **blue**

|  |  |  |  |
| --- | --- | --- | --- |
| piano | Co-occur1 | Co-occur2 | Co-occur3 |
|  | banjo | beethoven | banjo |
|  | bassoon | cello | bass |
|  | beethoven | clarinet | beethoven |
|  | cello | concerto | cello |
|  | clarinet | flute | clarinet |
|  | concerto | guitar | classical |
|  | continuo | handel | drum |
|  | flute | harp | flute |
|  | harp | mozart | guitar |
|  | harpsichord | oboe | harp |
|  | mandolin | orchestra | keyboard |
|  | oboe | recital | melodic |
|  | op | saxophone | mozart |
|  | quintet | soloist | professionally |
|  | sonata | sonata | sang |
|  | timpani | soprano | saxophone |
|  | trombone | trombone | strauss |
|  | viola | trumpet | tenor |
|  | violin | viola | viola |
|  | violoncello | violin | violin |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Co-occurrence-1 | | Co-occurrence-2 | | Co-occurrence-3 | |
| topic | semantic | Topic | Semantic | Topic | semantic |
| 7/20 | 13/20 | 8/20 | 12/20 | 10/20 | 10/20 |
| 35% | 65% | 40% | 60% | 50% | 50% |

Insights:

Most semantic-related words are musical-instruments.

Topic related words are words from the music-world, like Beethoven and Mozart as composers, and sonata and concerto.

1. brief description on implementation
   1. Estimation of the PMI values:

we count everything we need for PMI formula in "Association.py". The method that called "calc\_PMI()" in "VectorBuilder.py" use all the counters in Association object to calculate PMI. To save space we mapped each string (word or feature) to number, and save the number in the counters key.

we use strategy pattern to generalize the process, e.g.- the strategy builds the counters for Association object, for one of the co-occurrence type, and all the other method on those counters is generalize in Association object implementation. It should be noted that we filter the pairs like instructed (drop all words that found less than 100 times in the corpus), and also filter the features by selecting a fixed number of common features.

To keep the PMI valid after the filtering - we count only the pairs and the target word count before the filtering, and count the total pairs count and features count just to the remains words (while filtering). We used the identities presented in the presentation to make sure the PMI we calculate in valid after this steps.

* 1. Efficient algorithm for computing all similarities for a target word:

We start with computing all the vectors for all the remaining words (after the filtering). Each vector contains all the PMI results for the most common features for the target word. Afterwards we want to use this vector to compute similarities.

We implement similarities computing in two ways:

* **strict** – use cosine formula for two vectors words (method: VectorBuilder.cosine).
* **estimate** – using the estimate formula we learn in the class in iterative calculation on the vectors for given words and all the other vectors that we calculated in the first step (method: VectorBuilder.find\_similarities).

We instructed to **estimate** the similarities, so we use the estimate way in the end, and use the strict way to debug and check if the estimate calculation in the estimate way is valid (this tests doesn't delivered).

1. Word2vec:

Section-2:

The 20-most-common-words for each target word is in a separate file called  
"word2vec\_most\_common.csv".

* Conclusions for bow5:  
  similar words are at most (significantly) semantic-related words, we think it is because window of 5 contains more words that are semantically similar.  
    
  For example, for both car and piano 80% of words are semantic-related.
* Conclusions for deps:  
  similar words are significantly more semantic-related words.  
    
  For example, in both car and piano, there percentage is above 90.

Section-3:

The 20-most-common-contexts for each target word is in a separate file called  
"most\_common\_features\_word2vec.csv".

Comparison of Bow5:   
we can see that the features for each word matches the description of the similar words found for the target word, like horse for example, we can see that there are words like racing in different variance and the some of the similar words are race-horse,  
and for example the word bomb contains a lot of features that also found as similar words, that is due to the fact of this words appearing a lot in the same context.

Comparison of deps:   
we can see that the feauters for each word matches the description of the similar words found for the target word, like car that has features of model, designed and so on, so the similar words mostly used the same as car and will have similar dependency tree, like bike, jeep, types of specific car and so on.

Section-4:

For car:

Topic-related marked in **red**

Semantic-related marked in **blue**

|  |  |  |
| --- | --- | --- |
| car | Bow5 | deps |
|  | automobile | automobile |
|  | bike | bike |
|  | cars | cars |
|  | driver | jeep |
|  | front-engined | limo |
|  | limousine | limousine |
|  | lorry | lorry |
|  | mercedes-benz | minibus |
|  | mid-engined | minivan |
|  | minivan | motorcar |
|  | moped | motorcycle |
|  | motorbike | motorhome |
|  | motorcar | racecar |
|  | motorcycle | roadster |
|  | motorhome | speedboat |
|  | rear-engined | suv |
|  | suv | taxicab |
|  | three-wheeled | truck |
|  | truck | vehicle |
|  | vehicle | wagon |

|  |  |  |  |
| --- | --- | --- | --- |
| Bow5 | | deps | |
| topic | semantic | Topic | Semantic |
| 4/20 | 16/20 | 1/20 | 19/20 |
| 20% | 80% | 5% | 95% |

For piano:

Topic-related marked in **red**

Semantic-related marked in **blue**

|  |  |  |
| --- | --- | --- |
| piano | Bow5 | deps |
|  | accordion | accordion |
|  | bassoon | bassoon |
|  | cello | cello |
|  | clarinet | clarinet |
|  | concerto | clavichord |
|  | concertos | clavinet |
|  | flute | euphonium |
|  | harp | fortepiano |
|  | harpsichord | guitar |
|  | mandolin | harmonica |
|  | oboe | harpsichord |
|  | pianoforte | mandolin |
|  | saxophone | marimba |
|  | sonatas | pianoforte |
|  | trombone | saxophone |
|  | trumpet | trombone |
|  | vibraphone | trumpet |
|  | viola | vibraphone |
|  | violin | violin |
|  | violoncello | violoncello |

|  |  |  |  |
| --- | --- | --- | --- |
| Bow5 | | deps | |
| topic | semantic | Topic | Semantic |
| 4/20 | 16/20 | 0/20 | 20/20 |
| 20% | 80% | 0% | 100% |

In comparison of the word2vec-part to the previous part we can see that there are a lot of words for each of the target-words that has the same similar words in both cases.  
We can also notice that extending the window-size will cause to find for each word more semantic-related words then by a window of 2.  
One different is that in contrast to our opinion about deps in the first part (will find more topic-related words), in word2vec-part is appears that the dependency tree will cause to more semantic-related similar words.