An Extended Report on Similarity Score Calculation using Word2Vec Model for the extracted Final Glossary Set

1 Similarity Score Calculation using Word Embeddings

1.1 Some Prerequisites

- Used corpora
  - CrowdRE requirement specifications dataset (C_{CRE}) and
  - Wikipedia Home Automation category data for depth 2 (C_{HA}).
- Used word embedding model
  - Word2Vec by Google.
- Model parameter values (word2vec)
  - size = 100
  - window = 10
  - min_count = 1
  - model type (sg) = 1
  - workers = 4

  Note: The minimum count (min_count) parameter of the word2vec model is set to 1 for all the experiments in order to capture all the noun phrases irrespective of their frequencies. Further, the number of workers (workers) have been set to 4. The number of workers is same as the number of system cores, i.e. 4. It facilitates a faster training while generating the word vectors.
- Configuration Details
  - Programming Language : Python
  - Language Version : 3.7
  - Operating System : Windows 10
  - System Configuration : Intel Core-i5-7500 CPU, 4 GB DDR3 primary memory and a processor frequency of 3.40 GHz.

In this report, the (word) without any appended special characters denotes that the word belongs to Wikipedia home automation dataset (C_{HA}) whereas (_word_) specifies that it belongs to CrowdRE requirements (C_{CRE}) dataset. For example, 'access' belongs to C_{HA} dataset and '_access_' belongs to C_{CRE} dataset. Similarly, for the noun phrase 'blood pressure monitor', blood_pressure_monitor belongs to C_{HA} and _blood_pressure_monitor_ belongs to C_{CRE} dataset. The injection of special characters have been done to distinguish the same noun phrases in two different corpora.

The cosine similarity values are computed using the state-of-the-art neural word embeddings based word2vec model and the results are shown below alphabetically (noun phrases starting from A to Z). The noun phrases included in the final glossary set (304) are selected on the basis of semantic similarity scores (greater than or equal to 0.50). The computed semantic similarity scores are highlighted using magenta color text.

1.2 Results of the Word Embeddings for Noun Phrases Starting with "A"

1.2.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'access', w2 = '_access_')) : **0.5662872018676572**
2. print(model.wv.similarity(w1 = 'activity', w2 = '_activity_')) : **0.5179641158249992**
3. print(model.wv.similarity(w1 = 'adult', w2 = '_adult_')) : **0.6950987694161801**
4. print(model.wv.similarity(w1 = 'advance', w2 = '_advance_')) : **0.6490389441674601**
5. print(model.wv.similarity(w1 = 'air_conditioner', w2 = '_air_conditioner_')) : **0.804227330505443**
6. print(model.wv.similarity(w1 = 'air_conditioning', w2 = '_air_conditioning_')) : **0.6863136635706639**
7. print(model.wv.similarity(w1 = 'air_quality', w2 = '_air_quality_')) : **0.7576676220420514**
8. print(model.wv.similarity(w1 = 'alarm_clock', w2 = '_alarm_clock_')) : **0.6280880833873613**
9. print(model.wv.similarity(w1 = 'alert', w2 = '_alert_')) : **0.5130681948692939**
10. print(model.wv.similarity(w1 = 'amount', w2 = '_amount_')) : **0.6286442472141638**
11. print(model.wv.similarity(w1 = 'apartment', w2 = '_apartment_')) : **0.8519170122604073**
12. print(model.wv.similarity(w1 = 'app', w2 = '_app_')) : **0.532121592932453**
13. print(model.wv.similarity(w1 = 'appliance', w2 = '_appliance_')) : **0.5562555201773823**
14. print(model.wv.similarity(w1 = 'assistance', w2 = '_assistance_')) : **0.6994285822952181**
15. print(model.wv.similarity(w1 = 'amazon', w2 = '_amazon_')) : **0.5473458636365527**
16. print(model.wv.similarity(w1 = 'audio_system', w2 = '_audio_system_')) : **0.8517642704361398**
17. print(model.wv.similarity(w1 = 'automatic_door', w2 = '_automatic_door_')) : **0.9161588643512831**
18. print(model.wv.similarity(w1 = 'automobile', w2 = '_automobile_')) : **0.6164603873056764**

1.3 Results of the Word Embeddings for Noun Phrases Starting with "B"

1.3.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'baby', w2 = '_baby_')) : **0.6326568621770896**
2. print(model.wv.similarity(w1 = 'bag', w2 = '_bag_')) : **0.6165619760248173**
3. print(model.wv.similarity(w1 = 'base', w2 = '_base_')) : **0.5328936528614001**
4. print(model.wv.similarity(w1 = 'bath', w2 = '_bath_')) : **0.8519917297368664**
5. print(model.wv.similarity(w1 = 'bed', w2 = '_bed_')) : **0.5027916350371674**
6. print(model.wv.similarity(w1 = 'bill', w2 = '_bill_')) : **0.5906048145819094**
7. print(model.wv.similarity(w1 = 'blood_pressure', w2 = '_blood_pressure_')) : **0.8133013256438641**
8. print(model.wv.similarity(w1 = 'budget', w2 = '_budget_')) : **0.6227846082008428**
9. print(model.wv.similarity(w1 = 'business', w2 = '_business_')) : **0.5005154999179189**
10. print(model.wv.similarity(w1 = 'butt', w2 = '_butt_')) : **0.9307684337394448**
11. print(model.wv.similarity(w1 = 'bandwidth', w2 = '_bandwidth_')) : **0.5464401511192948**
12. print(model.wv.similarity(w1 = 'blood_pressure_monitor', w2 = '_blood_pressure_monitor_')) : **0.8091033405907111**
13. print(model.wv.similarity(w1 = 'body_temperature', w2 = '_body_temperature_')) : **0.5526216937471846**

1.4 Results of the Word Embeddings for Noun Phrases Starting with "C"

1.4.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'cabinet', w2 = '_cabinet_')) : **0.7283372354160973**
2. print(model.wv.similarity(w1 = 'car', w2 = '_car_')) : **0.5234939301382435**
3. print(model.wv.similarity(w1 = 'carbon', w2 = '_carbon_')) : **0.7215405390416327**
4. print(model.wv.similarity(w1 = 'carbon_monoxide', w2 = '_carbon_monoxide_')) : **0.7333106700226065**
5. print(model.wv.similarity(w1 = 'care', w2 = '_care_')) : **0.7008333927466854**
6. print(model.wv.similarity(w1 = 'carpet', w2 = '_carpet_')) : **0.8439354024843204**
7. print(model.wv.similarity(w1 = 'ceiling', w2 = '_ceiling_')) : **0.802168625863828**
8. print(model.wv.similarity(w1 = 'cell_phone', w2 = '_cell_phone_')) : **0.6797065645096653**
9. print(model.wv.similarity(w1 = 'chance', w2 = '_chance_')) : **0.8317216619997294**
10. print(model.wv.similarity(w1 = 'change', w2 = '_change_')) : **0.5137195250512419**
11. print(model.wv.similarity(w1 = 'child', w2 = '_child_')) : **0.6326651157534355**
12. print(model.wv.similarity(w1 = 'cleaning', w2 = '_cleaning_')) : **0.6793631542558116**
13. print(model.wv.similarity(w1 = 'closet', w2 = '_closet_')) : **0.86319994704556**
14. print(model.wv.similarity(w1 = 'closing', w2 = '_closing_')) : **0.829311597272123**
15. print(model.wv.similarity(w1 = 'clothing', w2 = '_clothing_')) : **0.8188251216839546**
16. print(model.wv.similarity(w1 = 'coffee', w2 = '_coffee_')) : **0.6610503637314492**
17. print(model.wv.similarity(w1 = 'coffee_pot', w2 = '_coffee_pot_')) : **0.9446839018243975**
18. print(model.wv.similarity(w1 = 'cold_air', w2 = '_cold_air_')) : **0.9519754959358602**
19. print(model.wv.similarity(w1 = 'comfort', w2 = '_comfort_')) : **0.7972376742996057**
20. print(model.wv.similarity(w1 = 'command', w2 = '_command_')) : **0.548872596073758**
21. print(model.wv.similarity(w1 = 'concentration', w2 = '_concentration_')) : **0.842701997066317**
22. print(model.wv.similarity(w1 = 'conserve_water', w2 = '_conserve_water_')) : **0.9103232916983944**
23. print(model.wv.similarity(w1 = 'cook', w2 = '_cook_')) : **0.6386050818835453**
24. print(model.wv.similarity(w1 = 'cooking', w2 = '_cooking_')) : **0.561803636387531**
25. print(model.wv.similarity(w1 = 'cooler', w2 = '_cooler_')) : **0.6583985702195873**
26. print(model.wv.similarity(w1 = 'cooling', w2 = '_cooling_')) : **0.6980218646929879**
27. print(model.wv.similarity(w1 = 'carbon_monoxide_detector', w2 = '_carbon_monoxide_detector_')) : **0.7774689635107982**
28. print(model.wv.similarity(w1 = 'cell', w2 = '_cell_')) : **0.5891817205261414**
29. print(model.wv.similarity(w1 = 'comfortable temperature', w2 = '_comfortable_temperature_')) : **0.9418016679628247**
30. print(model.wv.similarity(w1 = 'cpu', w2 = '_cpu_')) : **0.5308281989125421**

1.5 Results of the Word Embeddings for Noun Phrases Starting with "D"

1.5.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'damage', w2 = '_damage_')) : **0.7499398940404307**
2. print(model.wv.similarity(w1 = 'danger', w2 = '_danger_')) : **0.7371222327887585**
3. print(model.wv.similarity(w1 = 'dark', w2 = '_dark_')) : **0.6698054255002639**
4. print(model.wv.similarity(w1 = 'day', w2 = '_day_')) : **0.523601338005479**
5. print(model.wv.similarity(w1 = 'demand', w2 = '_demand_')) : **0.5883513824830934**
6. print(model.wv.similarity(w1 = 'desired_temperature', w2 = '_desired_temperature_')) : **0.9186766955821704**
7. print(model.wv.similarity(w1 = 'detergent', w2 = '_detergent_')) : **0.8911856480763459**
8. print(model.wv.similarity(w1 = 'direction', w2 = '_direction_')) : **0.5206562492615674**
9. print(model.wv.similarity(w1 = 'dirt', w2 = '_dirt_')) : **0.7397702673963458**
10. print(model.wv.similarity(w1 = 'dishwasher', w2 = '_dishwasher_')) : **0.6959458790923763**
11. print(model.wv.similarity(w1 = 'dog', w2 = '_dog_')) : **0.5618287422407702**
12. print(model.wv.similarity(w1 = 'door', w2 = '_door_')) : **0.6498060157235337**
13. print(model.wv.similarity(w1 = 'doorbell', w2 = '_doorbell_')) : **0.7228514267464352**
14. print(model.wv.similarity(w1 = 'dryer', w2 = '_dryer_')) : **0.7472215001293916**
15. print(model.wv.similarity(w1 = 'dust', w2 = '_dust_')) : **0.607723810912314**
16. print(model.wv.similarity(w1 = 'database', w2 = '_database_')) : **0.5273640840359524**

1.6 Results of the Word Embeddings for Noun Phrases Starting with "E"

1.6.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'effort', w2 = '_effort_')) : **0.5023131341569708**
2. print(model.wv.similarity(w1 = 'electricity', w2 = '_electricity_')) : **0.6921177259758948**
3. print(model.wv.similarity(w1 = 'email', w2 = '_email_')) : **0.6608218129131527**
4. print(model.wv.similarity(w1 = 'emergency', w2 = '_emergency_')) : **0.7402727573597694**
5. print(model.wv.similarity(w1 = 'energy', w2 = '_energy_')) : **0.5323848628149129**
6. print(model.wv.similarity(w1 = 'energy_consumption', w2 = '_energy_consumption_')) : **0.7771752873955737**
7. print(model.wv.similarity(w1 = 'energy_efficient', w2 = '_energy_efficient_')) : **0.7637368584779861**
8. print(model.wv.similarity(w1 = 'energy_usage', w2 = '_energy_usage_')) : **0.827957013248923**
9. print(model.wv.similarity(w1 = 'enter', w2 = '_enter_')) : **0.6691496895982869**
10. print(model.wv.similarity(w1 = 'entertainment', w2 = '_entertainment_')) : **0.6620125735989243**
11. print(model.wv.similarity(w1 = 'entertainment_system', w2 = '_entertainment_system_')) : **0.855363383911214**
12. print(model.wv.similarity(w1 = 'entry', w2 = '_entry_')) : **0.570370050835042**
13. print(model.wv.similarity(w1 = 'example', w2 = '_example_')) : **0.5708812020744881**
14. print(model.wv.similarity(w1 = 'excess', w2 = '_excess_')) : **0.7946063803833303**
15. print(model.wv.similarity(w1 = 'excess_moisture', w2 = '_excess_moisture_')) : **0.9389538398350743**
16. print(model.wv.similarity(w1 = 'exercise', w2 = '_exercise_')) : **0.7878707556588451**
17. print(model.wv.similarity(w1 = 'experience', w2 = '_experience_')) : **0.5853260558687032**
18. print(model.wv.similarity(w1 = 'electric_blanket', w2 = '_electric_blanket_')) : **0.9722460444561772**

1.7 Results of the Word Embeddings for Noun Phrases Starting with "F"

1.7.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'face_detection', w2 = '_face_detection_')) : **0.6668423065247553**
2. print(model.wv.similarity(w1 = 'facial_recognition', w2 = '_facial_recognition_')) : **0.6327893119146906**
3. print(model.wv.similarity(w1 = 'fingerprint_scanner', w2 = '_fingerprint_scanner_')) : **0.6735032648218964**
4. print(model.wv.similarity(w1 = 'fingerprint_sensor', w2 = '_fingerprint_sensor_')) : **0.7015982635291472**
5. print(model.wv.similarity(w1 = 'fitbit', w2 = '_fitbit_')) : **0.6585077071063425**
6. print(model.wv.similarity(w1 = 'face', w2 = '_face_')) : **0.6435585894438383**
7. print(model.wv.similarity(w1 = 'failure', w2 = '_failure_')) : **0.7516540361771639**
8. print(model.wv.similarity(w1 = 'fire', w2 = '_fire_')) : **0.5210947696816113**
9. print(model.wv.similarity(w1 = 'fitness', w2 = '_fitness_')) : **0.5656172573147116**
10. print(model.wv.similarity(w1 = 'floor', w2 = '_floor_')) : **0.6970938262571433**
11. print(model.wv.similarity(w1 = 'food', w2 = '_food_')) : **0.6086799106816846**
12. print(model.wv.similarity(w1 = 'fresh_air', w2 = '_fresh_air_')) : **0.8996430684623354**
13. print(model.wv.similarity(w1 = 'fridge', w2 = '_fridge_')) : **0.7122457444698929**
14. print(model.wv.similarity(w1 = 'front', w2 = '_front_')) : **0.5008606883298259**
15. print(model.wv.similarity(w1 = 'front_door', w2 = '_front_door_')) : **0.7854621035245314**
16. print(model.wv.similarity(w1 = 'fun', w2 = '_fun_')) : **0.8410425267050566**
17. print(model.wv.similarity(w1 = 'furnace', w2 = '_furnace_')) : **0.655274328466265**

1.8 Results of the Word Embeddings for Noun Phrases Starting with "G"

1.8.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'game', w2 = '_game_')) : **0.5902164483255778**
2. print(model.wv.similarity(w1 = 'garage', w2 = '_garage_')) : **0.7353258731072628**
3. print(model.wv.similarity(w1 = 'garage_door', w2 = '_garage_door_')) : **0.6879694955549169**
4. print(model.wv.similarity(w1 = 'garden', w2 = '_garden_')) : **0.8153134348993005**
5. print(model.wv.similarity(w1 = 'gas', w2 = '_gas_')) : **0.5743098235312017**
6. print(model.wv.similarity(w1 = 'gps_location', w2 = '_gps_location_')) : **0.9102094852295284**
7. print(model.wv.similarity(w1 = 'grass', w2 = '_grass_')) : **0.915309582633519**
8. print(model.wv.similarity(w1 = 'ground', w2 = '_ground_')) : **0.6976356910601693**
9. print(model.wv.similarity(w1 = 'geyser', w2 = '_geyser_')) : **0.8969245660612644**

1.9 Results of the Word Embeddings for Noun Phrases Starting with "H"

1.9.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'hair', w2 = '_hair_')) : **0.8665502055113242**
2. print(model.wv.similarity(w1 = 'hand', w2 = '_hand_')) : **0.7341568750342076**
3. print(model.wv.similarity(w1 = 'head', w2 = '_head_')) : **0.6746422010640829**
4. print(model.wv.similarity(w1 = 'health', w2 = '_health_')) : **0.5791338891524334**
5. print(model.wv.similarity(w1 = 'heater', w2 = '_heater_')) : **0.5766897714661732**
6. print(model.wv.similarity(w1 = 'heating', w2 = '_heating_')) : **0.6244167227964039**
7. print(model.wv.similarity(w1 = 'help', w2 = '_help_')) : **0.6382439343650044**
8. print(model.wv.similarity(w1 = 'hot_water', w2 = '_hot_water_')) : **0.6501157517972309**
9. print(model.wv.similarity(w1 = 'hour', w2 = '_hour_')) : **0.5355431960763423**
10. print(model.wv.similarity(w1 = 'house', w2 = '_house_')) : **0.5358512996323941**
11. print(model.wv.similarity(w1 = 'household', w2 = '_household_')) : **0.6626324466406348**
12. print(model.wv.similarity(w1 = 'humidity', w2 = '_humidity_')) : **0.8157118982894879**
13. print(model.wv.similarity(w1 = 'hurt', w2 = '_hurt_')) : **0.8058665527239088**
14. print(model.wv.similarity(w1 = 'heart_rate', w2 = '_heart_rate_')) : **0.6172836221412384**
15. print(model.wv.similarity(w1 = 'high_resolution', w2 = '_high_resolution_')) : **0.7692799662358001**
16. print(model.wv.similarity(w1 = 'hologram', w2 = '_hologram_')) : **0.9035501540664591**

1.10 Results of the Word Embeddings for Noun Phrases Starting with "I"

1.10.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'ice', w2 = '_ice_')) : **0.8111446456009674**
2. print(model.wv.similarity(w1 = 'ice_cream', w2 = '_ice_cream_')) : **0.5765428096491638**
3. print(model.wv.similarity(w1 = 'ideal_temperature', w2 = '_ideal_temperature_')) : **0.8345078863557374**
4. print(model.wv.similarity(w1 = 'indoor', w2 = '_indoor_')) : **0.5985697641490322**
5. print(model.wv.similarity(w1 = 'insulation', w2 = '_insulation_')) : **0.7468542639966649**
6. print(model.wv.similarity(w1 = 'interest', w2 = '_interest_')) : **0.5737046430556474**
7. print(model.wv.similarity(w1 = 'inventory', w2 = '_inventory_')) : **0.7200715637655559**
8. print(model.wv.similarity(w1 = 'iphone', w2 = '_iphone_')) : **0.5018294300759831**
9. print(model.wv.similarity(w1 = 'internet_service', w2 = '_internet_service_')) : **0.7679102873365768**
10. print(model.wv.similarity(w1 = 'ipad', w2 = '_ipad_')) : **0.6262225011416439**

1.11 Results of the Word Embeddings for Noun Phrases Starting with "K"

1.11.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'kitchen', w2 = '_kitchen_')) : **0.5648027566794277**

1.12 Results of the Word Embeddings for Noun Phrases Starting with "L"

1.12.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'laptop', w2 = '_laptop_')) : **0.5698668695195921**
2. print(model.wv.similarity(w1 = 'laser', w2 = '_laser_')) : **0.6032596038680393**
3. print(model.wv.similarity(w1 = 'laundry', w2 = '_laundry_')) : **0.7694020070796255**
4. print(model.wv.similarity(w1 = 'lawn', w2 = '_lawn_')) : **0.6199319263453966**
5. print(model.wv.similarity(w1 = 'life', w2 = '_life_')) : **0.5134512421763047**
6. print(model.wv.similarity(w1 = 'light', w2 = '_light_')) : **0.5992129174551024**
7. print(model.wv.similarity(w1 = 'lighting', w2 = '_lighting_')) : **0.555517219228063**
8. print(model.wv.similarity(w1 = 'location', w2 = '_location_')) : **0.5501864110011101**
9. print(model.wv.similarity(w1 = 'lock', w2 = '_lock_')) : **0.6579092017022701**

1.13 Results of the Word Embeddings for Noun Phrases Starting with "M"

1.13.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'microphone', w2 = '_microphone_')) : **0.5954789957441267**
2. print(model.wv.similarity(w1 = 'mobile_device', w2 = '_mobile_device_')) : **0.531668498369388**
3. print(model.wv.similarity(w1 = 'monitoring_system', w2 = '_monitoring_system_')) : **0.7698533780323409**
4. print(model.wv.similarity(w1 = 'music_player', w2 = '_music_player_')) : **0.6186559038301996**
5. print(model.wv.similarity(w1 = 'mail', w2 = '_mail_')) : **0.5188367875259746**
6. print(model.wv.similarity(w1 = 'maintenance', w2 = '_maintenance_')) : **0.6493115501091078**
7. print(model.wv.similarity(w1 = 'medication', w2 = '_medication_')) : **0.7991858039435427**
8. print(model.wv.similarity(w1 = 'message', w2 = '_message_')) : **0.6021174327254846**
9. print(model.wv.similarity(w1 = 'milk', w2 = '_milk_')) : **0.8885902875274797**
10. print(model.wv.similarity(w1 = 'mind', w2 = '_mind_')) : **0.785838492949404**
11. print(model.wv.similarity(w1 = 'minute', w2 = '_minute_')) : **0.6002040008964247**
12. print(model.wv.similarity(w1 = 'mistake', w2 = '_mistake_')) : **0.8736302366512214**
13. print(model.wv.similarity(w1 = 'moisture', w2 = '_moisture_')) : **0.8832620641890285**
14. print(model.wv.similarity(w1 = 'money', w2 = '_money_')) : **0.5512359691608977**
15. print(model.wv.similarity(w1 = 'morning', w2 = '_morning_')) : **0.8208791327568099**
16. print(model.wv.similarity(w1 = 'motion', w2 = '_motion_')) : **0.6473464085078947**
17. print(model.wv.similarity(w1 = 'motion_sensor', w2 = '_motion_sensor_')) : **0.6606161962430533**
18. print(model.wv.similarity(w1 = 'mouth', w2 = '_mouth_')) : **0.8215846635444233**
19. print(model.wv.similarity(w1 = 'movement', w2 = '_movement_')) : **0.7262213498252901**
20. print(model.wv.similarity(w1 = 'movie', w2 = '_movie_')) : **0.740988041390475**
21. print(model.wv.similarity(w1 = 'music', w2 = '_music_')) : **0.6542632971169604**
22. print(model.wv.similarity(w1 = 'music_system', w2 = '_music_system_')) : **0.8911974205922891**

1.14 Results of the Word Embeddings for Noun Phrases Starting with "N"

1.14.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'netflix', w2 = '_netflix_')) : **0.6974084921934927**
2. print(model.wv.similarity(w1 = 'news', w2 = '_news_')) : **0.5832196588565749**
3. print(model.wv.similarity(w1 = 'night', w2 = '_night_')) : **0.6377917237546893**
4. print(model.wv.similarity(w1 = 'noise', w2 = '_noise_')) : **0.6392215949188957**
5. print(model.wv.similarity(w1 = 'notice', w2 = '_notice_')) : **0.774515397706896**
6. print(model.wv.similarity(w1 = 'notification', w2 = '_notification_')) : **0.5936643904733907**

1.15 Results of the Word Embeddings for Noun Phrases Starting with "O"

1.15.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'office', w2 = '_office_')) : **0.5168876670843274**
2. print(model.wv.similarity(w1 = 'order', w2 = '_order_')) : **0.5464126132598501**
3. print(model.wv.similarity(w1 = 'outdoor_motion', w2 = '_outdoor_motion_')) : **0.9062187991044891**

1.16 Results of the Word Embeddings for Noun Phrases Starting with "P"

1.16.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'personal_computer', w2 = '_personal_computer_')) : **0.6067280984554811**
2. print(model.wv.similarity(w1 = 'permission', w2 = '_permission_')) : **0.7274804521314394**
3. print(model.wv.similarity(w1 = 'person', w2 = '_person_')) : **0.7289492637887277**
4. print(model.wv.similarity(w1 = 'pet', w2 = '_pet_')) : **0.57764754031565**
5. print(model.wv.similarity(w1 = 'picture', w2 = '_picture_')) : **0.6355350018318429**
6. print(model.wv.similarity(w1 = 'pizza', w2 = '_pizza_')) : **0.9116165367596991**
7. print(model.wv.similarity(w1 = 'place', w2 = '_place_')) : **0.5755754451617269**
8. print(model.wv.similarity(w1 = 'pm', w2 = '_pm_')) : **0.7897892157298084**
9. print(model.wv.similarity(w1 = 'pollution', w2 = '_pollution_')) : **0.7173926420107305**
10. print(model.wv.similarity(w1 = 'pool', w2 = '_pool_')) : **0.532385465731591**
11. print(model.wv.similarity(w1 = 'power', w2 = '_power_')) : **0.5541164880413066**
12. print(model.wv.similarity(w1 = 'power_consumption', w2 = '_power_consumption_')) : **0.7094979654587958**
13. print(model.wv.similarity(w1 = 'power_usage', w2 = '_power_usage_')) : **0.8957814180543909**
14. print(model.wv.similarity(w1 = 'practice', w2 = '_practice_')) : **0.5932601342735557**
15. print(model.wv.similarity(w1 = 'preferred_temperature', w2 = '_preferred_temperature_')) : **0.9486107261057719**
16. print(model.wv.similarity(w1 = 'presence', w2 = '_presence_')) : **0.6770143545588111**
17. print(model.wv.similarity(w1 = 'pressure', w2 = '_pressure_')) : **0.6701778973392047**
18. print(model.wv.similarity(w1 = 'prevent', w2 = '_prevent_')) : **0.6786624078312884**
19. print(model.wv.similarity(w1 = 'purpose', w2 = '_purpose_')) : **0.6007623145004317**

1.17 Results of the Word Embeddings for Noun Phrases Starting with "Q"

1.17.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'quality', w2 = '_quality_')) : **0.5574427584146255**

1.18 Results of the Word Embeddings for Noun Phrases Starting with "R"

1.18.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'radiator', w2 = '_radiator_')) : **0.6018269851535123**
2. print(model.wv.similarity(w1 = 'remote_control', w2 = '_remote_control_')) : **0.5544773938562103**
3. print(model.wv.similarity(w1 = 'rfid_chip', w2 = '_rfid_chip_')) : **0.533982161401331**
4. print(model.wv.similarity(w1 = 'robotic_vacuum', w2 = '_robotic_vacuum_')) : **0.6478055929529478**
5. print(model.wv.similarity(w1 = 'real_time', w2 = '_real_time_')) : **0.7243460336179102**
6. print(model.wv.similarity(w1 = 'recognition', w2 = '_recognition_')) : **0.5524759113609652**
7. print(model.wv.similarity(w1 = 'recommend', w2 = '_recommend_')) : **0.852423983860642**
8. print(model.wv.similarity(w1 = 'record', w2 = '_record_')) : **0.5419962451371658**
9. print(model.wv.similarity(w1 = 'refrigerator', w2 = '_refrigerator_')) : **0.6457153059522223**
10. print(model.wv.similarity(w1 = 'remind', w2 = '_remind_')) : **0.8005435722216818**
11. print(model.wv.similarity(w1 = 'remote', w2 = '_remote_')) : **0.5113956395051871**
12. print(model.wv.similarity(w1 = 'remote_access', w2 = '_remote_access_')) : **0.6144694626450362**
13. print(model.wv.similarity(w1 = 'repair', w2 = '_repair_')) : **0.8286567577042714**
14. print(model.wv.similarity(w1 = 'respond', w2 = '_respond_')) : **0.6542746003042242**
15. print(model.wv.similarity(w1 = 'room', w2 = '_room_')) : **0.5839847201966661**
16. print(model.wv.similarity(w1 = 'room_temperature', w2 = '_room_temperature_')) : **0.8387639994045268**
17. print(model.wv.similarity(w1 = 'room_thermostat', w2 = '_room_thermostat_')) : **0.8757631698955558**

1.19 Results of the Word Embeddings for Noun Phrases Starting with "S"

1.19.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'safety_system', w2 = '_safety_system_')) : **0.9091600103047455**
2. print(model.wv.similarity(w1 = 'security_camera', w2 = '_security_camera_')) : **0.6315117561263749**
3. print(model.wv.similarity(w1 = 'siri', w2 = '_siri_')) : **0.7186028842496186**
4. print(model.wv.similarity(w1 = 'smart_alarm_clock', w2 = '_smart_alarm_clock_')) : **0.9534728231129911**
5. print(model.wv.similarity(w1 = 'smart_card', w2 = '_smart_card_')) : **0.5502507547577786**
6. print(model.wv.similarity(w1 = 'smart_key', w2 = '_smart_key')) : **0.7620208830131356**
7. print(model.wv.similarity(w1 = 'smart_light', w2 = '_smart_light_')) : **0.9081975431312108**
8. print(model.wv.similarity(w1 = 'smart_sensor', w2 = '_smart_sensor_')) : **0.8645057937978999**
9. print(model.wv.similarity(w1 = 'smart_tag', w2 = '_smart_tag_')) : **0.7437252538008385**

10. print(model.wv.similarity(w1 = 'smart_water', w2 = '_smart_water_')) : **0.8521815418488502**
11. print(model.wv.similarity(w1 = 'solar_panel', w2 = '_solar_panel_')) : **0.7435543610646698**
12. print(model.wv.similarity(w1 = 'solar_roof', w2 = '_solar_roof_')) : **0.9687508975729935**
13. print(model.wv.similarity(w1 = 'solar_system', w2 = '_solar_system_')) : **0.8994893798203093**
14. print(model.wv.similarity(w1 = 'surround_sound', w2 = '_surround_sound_')) : **0.6137795052184909**
15. print(model.wv.similarity(w1 = 'safety', w2 = '_safety_')) : **0.5554991621533677**
16. print(model.wv.similarity(w1 = 'sale', w2 = '_sale_')) : **0.5459978296356415**
17. print(model.wv.similarity(w1 = 'scale', w2 = '_scale_')) : **0.5063918515504089**
18. print(model.wv.similarity(w1 = 'schedule', w2 = '_schedule_')) : **0.6829312200214028**
19. print(model.wv.similarity(w1 = 'school', w2 = '_school_')) : **0.5287600630257405**
20. print(model.wv.similarity(w1 = 'screen', w2 = '_screen_')) : **0.5346685010092564**
21. print(model.wv.similarity(w1 = 'security_system', w2 = '_security_system_')) : **0.6322845911459222**
22. print(model.wv.similarity(w1 = 'shape', w2 = '_shape_')) : **0.6013891518182042**
23. print(model.wv.similarity(w1 = 'shower', w2 = '_shower_')) : **0.8274807477607332**
24. print(model.wv.similarity(w1 = 'sleep', w2 = '_sleep_')) : **0.7858015092011609**
25. print(model.wv.similarity(w1 = 'sleeping', w2 = '_sleeping_')) : **0.891041322639315**
26. print(model.wv.similarity(w1 = 'smart_device', w2 = '_smart_device_')) : **0.5897182892325411**
27. print(model.wv.similarity(w1 = 'smart_fridge', w2 = '_smart_fridge_')) : **0.9286014141456842**
28. print(model.wv.similarity(w1 = 'smart_tv', w2 = '_smart_tv_')) : **0.5652349078599798**
29. print(model.wv.similarity(w1 = 'smoke', w2 = '_smoke_')) : **0.7161139576164125**
30. print(model.wv.similarity(w1 = 'soap', w2 = '_soap_')) : **0.9198978523761443**
31. print(model.wv.similarity(w1 = 'song', w2 = '_song_')) : **0.7028919451480149**
32. print(model.wv.similarity(w1 = 'sound', w2 = '_sound_')) : **0.5192324049021264**
33. print(model.wv.similarity(w1 = 'speed', w2 = '_speed_')) : **0.5888094045190705**
34. print(model.wv.similarity(w1 = 'steam', w2 = '_steam_')) : **0.599097260824721**
35. print(model.wv.similarity(w1 = 'step', w2 = '_step_')) : **0.5301336258176286**
36. print(model.wv.similarity(w1 = 'stereo', w2 = '_stereo_')) : **0.5619876509451549**
37. print(model.wv.similarity(w1 = 'stock', w2 = '_stock_')) : **0.5247983534533636**
38. print(model.wv.similarity(w1 = 'stove', w2 = '_stove_')) : **0.5078374099871286**
39. print(model.wv.similarity(w1 = 'stress', w2 = '_stress_')) : **0.6926370379262998**
40. print(model.wv.similarity(w1 = 'stuff', w2 = '_stuff_')) : **0.6984331807762981**
41. print(model.wv.similarity(w1 = 'summer', w2 = '_summer_')) : **0.6705512028710653**
42. print(model.wv.similarity(w1 = 'sunlight', w2 = '_sunlight_')) : **0.8475387988188792**

1.20 Results of the Word Embeddings for Noun Phrases Starting with "T"

1.20.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'thermometer', w2 = '_thermometer_')) : **0.6968659488543418**
2. print(model.wv.similarity(w1 = 'tab', w2 = '_tab_')) : **0.6511451243489557**
3. print(model.wv.similarity(w1 = 'tank', w2 = '_tank_')) : **0.558375939421357**
4. print(model.wv.similarity(w1 = 'temperature', w2 = '_temperature_')) : **0.5735258788782255**
5. print(model.wv.similarity(w1 = 'text', w2 = '_text_')) : **0.5506511080901442**
6. print(model.wv.similarity(w1 = 'text_message', w2 = '_text_message_')) : **0.7600195691020499**
7. print(model.wv.similarity(w1 = 'theft', w2 = '_theft_')) : **0.5204177589000905**
8. print(model.wv.similarity(w1 = 'thermostat', w2 = '_thermostat_')) : **0.6345073103885303**
9. print(model.wv.similarity(w1 = 'track', w2 = '_track_')) : **0.5243313189958458**
10. print(model.wv.similarity(w1 = 'traffic', w2 = '_traffic_')) : **0.5963274602336464**
11. print(model.wv.similarity(w1 = 'trip', w2 = '_trip_')) : **0.7129693036347393**
12. print(model.wv.similarity(w1 = 'tub', w2 = '_tub_')) : **0.7919586853321345**
13. print(model.wv.similarity(w1 = 'turn', w2 = '_turn_')) : **0.5506338335971104**

1.21 Results of the Word Embeddings for Noun Phrases Starting with "U"

1.21.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'usage', w2 = '_usage_')) : **0.6533943369204125**
2. print(model.wv.similarity(w1 = 'utility', w2 = '_utility_')) : **0.5091110094930764**

1.22 Results of the Word Embeddings for Noun Phrases Starting with "V"

1.22.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'vacation', w2 = '_vacation_')) : **0.952014857724238**
2. print(model.wv.similarity(w1 = 'vacuum', w2 = '_vacuum_')) : **0.6974102000144928**
3. print(model.wv.similarity(w1 = 'vehicle', w2 = '_vehicle_')) : **0.514511828148539**
4. print(model.wv.similarity(w1 = 'visitor', w2 = '_visitor_')) : **0.7721771370113777**
5. print(model.wv.similarity(w1 = 'voice', w2 = '_voice_')) : **0.6429354282205574**
6. print(model.wv.similarity(w1 = 'voice_recognition', w2 = '_voice_recognition_')) : **0.7602848234824817**
7. print(model.wv.similarity(w1 = 'volume', w2 = '_volume_')) : **0.5221348427176818**
8. print(model.wv.similarity(w1 = 'voice_control', w2 = '_voice_control_')) : **0.6737586759469159**

1.23 Results of the Word Embeddings for Noun Phrases Starting with "W"

1.23.1 Similarity Scores:

1. print(model.wv.similarity(w1 = 'wall', w2 = '_wall_')) : **0.5794485866615926**
2. print(model.wv.similarity(w1 = 'washer', w2 = '_washer_')) : **0.6968208642475708**
3. print(model.wv.similarity(w1 = 'waste', w2 = '_waste_')) : **0.5387957354758203**
4. print(model.wv.similarity(w1 = 'waste_energy', w2 = '_waste_energy_')) : **0.9021811803136865**
5. print(model.wv.similarity(w1 = 'water', w2 = '_water_')) : **0.5287057937598258**
6. print(model.wv.similarity(w1 = 'water_damage', w2 = '_water_damage_')) : **0.9348746668206385**
7. print(model.wv.similarity(w1 = 'water_usage', w2 = '_water_usage_')) : **0.9627634975652943**
8. print(model.wv.similarity(w1 = 'weather', w2 = '_weather_')) : **0.7036245103173182**
9. print(model.wv.similarity(w1 = 'week', w2 = '_week_')) : **0.6633181950663879**
10. print(model.wv.similarity(w1 = 'weight', w2 = '_weight_')) : **0.5363308299589389**
11. print(model.wv.similarity(w1 = 'wifi', w2 = '_wifi_')) : **0.5842317527563429**
12. print(model.wv.similarity(w1 = 'wireless', w2 = '_wireless_')) : **0.5107122567745821**
13. print(model.wv.similarity(w1 = 'worry', w2 = '_worry_')) : **0.7592801055129546**
14. print(model.wv.similarity(w1 = 'wireless_speaker', w2 = '_wireless_speaker_')) : **0.7956386085148504**