9. Semantic similarity and Synonymy relation

Joonas Soudunsaari, Subham Chauhan, Anandharuban Panchanathan

Project on github: https://github.com/kannadan/NLP-semantic-similarity

1 Introduction

Semantic Similarity is a metric defined as likeness between words based on their semantic content or closeness of their meaning. It is used in natural language processing, information retrieval, information extraction, etc (Meng et al. 2012). Semantic similarity implements a valuable tool to identify, analyze and implement pattern matching and text snipping.

A large dataset can be organized and classified based on their semantic content which is an important aspect of data mining and other computer aspects. For example, we can take two sentences with semantic content like 'I like to watch television' and 'I like to read books.' These both sentences imply that the person likes to watch television and to read books. So, calculating the semantic similarity of these sentences can help machines understand the closeness and similarity of certain texts.

Various approaches are used to assess the semantic similarities i.e., Wu-Palmer's similarity, Cosine similarity and Lin similarities. Since the relatedness of the words are different based on the approach that has been used, it is important to compare the results obtained from different approaches (Wicaksana et al. 2005).

1.1 Problem description

This project aims at investigating the semantic similarities between the selected pairs of synonyms and antonyms. This study utilizes the Wu-Palmer similarity, cosine similarity and lin similarity. The WordNet, Word2Vec vector, Glove embedding, and Duos are used in this study for word selection. The selected similarities were assessed based on the average similarities, standard deviations. The popularity of the words was assessed using brown corpus and Pearson coefficient were used to compare the similarities based on the popularity.

Some Semantic Similarity techniques:

Wu-Palmer's Similarity

Wu-Palmer's Similarity measure calculates closeness by considering the depths of the two synsets in WordNet Taxonomies, along with the depth of least common subsumer. The result can be anywhere greater than 0 or less than or equals to zero (0<Similarity<=1)

$$\mathbf{Wu - Palmer} = 2 * \frac{\mathbf{depth}\,(\mathbf{lcs}(\mathbf{s1}, \mathbf{s2}))}{(\mathbf{depth}\,(\mathbf{s1}) + \mathbf{depth}\,(\mathbf{s2}))}$$

Cosine Similarity

Cosine Similarity is another semantic similarity method that determines how similar the data objects are by using datasets as a vector. The result is calculated by multiplying the two vectors and dividing it by the product of the length of the vectors.

$$\mathrm{Cos}(x, y) = x \cdot y / ||x|| * ||y||$$

Lin similarity

Lin similarity measures the similarity between concepts belonging to concept classes rather than similarities between them.

$$Sim_{lin}(c_1, c_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

2 Methodology

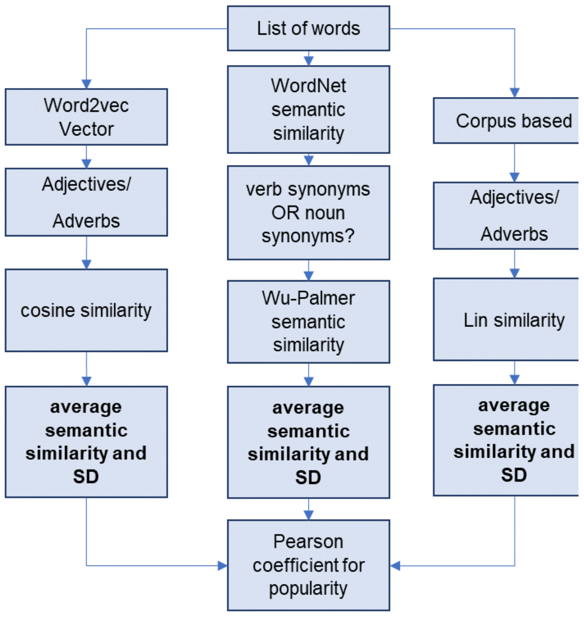

List of words
Word2vec Vector
WordNet semantic similarity
Corpus based
Adjectives/ Adverbs
verb synonyms OR noun synonyms?
Adjectives/ Adverbs
cosine similarity
Wu-Palmer semantic similarity
Lin similarity
average semantic similarity and SD
average semantic similarity and SD
average semantic similarity and SD
Pearson coefficient for popularity

Figure 1. Methodology

A) Word processing and results

To get the synonyms into a form that our python script could handle, they needed some pre-processing. To get 96 words from the web site, they need to be manually copy pasted to txt file. This luckily resulted in the words and their list of synonyms being all in one line each, where the main word and its synonyms were separated by a dash. The dash was some special Unicode, but text editor was used to quickly swap them to standard dash. This was important for word processing, as the dash was used to split the words groups, and the odd dash could not be recognized by the program. Some words like Say/Tell had two options for main word, which would have caused some differentiation compared to the others, so a decision was made to eliminate the second main word from these cases.

After the text file was prepped, the main work fell to the script to read every word into variables to use. Basic pythonic methods like splits and strips were used to handle this part. The only real issue was to remove the line ending that would have made the last synonym on each line unrecognizable to the similarity algorithms. Similar process was used for later parts of the tasks where we analyze antonyms and duos, but these were much simpler since all words had only their opposite, instead of a list of corresponding synonyms.

All the separate tasks result in a list of words and their similarity values, so for analysis and further processing if needed, the results are saved in csv files, usually one per task. These will be structured in simple columns where the main word is in the first column, synonyms/antonyms/duos on the second and statistics in following columns. This same model will be used in all tasks, and this will allow simple reading of the results and creating graphs as needed.

B) Synonym similarity with WordNet

The first part of the investigation is to calculate semantic similarity of synonyms on our list using wordnet as our similarity measurement as our engine. For this first part, only nouns and verbs are used as adjectives and adverbs do not have hierarchical representation in wordnet. In practice, we took a synset from the word using python nltk library with additional parameters of verb and noun. If both synset requests returned empty list, we knew the word was unusable. Out of the 96 synonyms on our list, we managed to make a comparison of 77 synonyms. The average similarity score varied a lot for synonym groups but was generally on the low side. Minimum average was 0.117 and maximum 0.831. The average for these 77-word groups was 0.352 and as the maximum value for wordnet similarity is 1, these scores are rather low. Similarly standard distribution had an average of 0.13 with a minimum of 0.009 and maximum of 0.372.

Anger	Nettle(0.11)	0.11
Answer	Reply(1.0) Retort(0.28)…	0.64
ask	Question(0.12) Demand(0.16)…	0.4

Table 1 Result example of wordnet similarity (word, synonym, average)

C) Synonym similarity with word2vec

Calculating semantic similarity using word2vec method is the second part of this investigation. The process is similar to wordnet abroach, except that word2vec can in fact be used to calculate similarity between any word, adverbs included. As each word receives a vector representation and then we calculate the similarity using cosine distance of these vectors, the part of speech does not matter. The results were saved in a similar format as wordnet similarity task. Minimum average value for synonym group was 0,943 and maximum was 1,045 whereas total average is 0.998. Standard deviation average is 0.08 with minimum of 0,058 disqualifying zero values when there was only 1 comparison and maximum of 0,124.

D) Adjective/adverb similarity

In part 1 we calculated similarity using word net and wu-palmer similarity. We had to discount adverbs and adjective as they had no hierarchical representation I synset models. There however is a workaround that we will test. In order to get a usable word representation, we will take the trouble words and look for derivationally similar word and do comparison calculations using that replacement word. Examples of derivative words are angry -> anger, awful -> awfulness. After finding these replacement words, we follow our previous baths to calculate wu-palmer similarity from wordnet and cosine distance from word2vec methods. In addition to these, we also used lin similarity to calculate additional comparison value. The derivational words were found using word net and its lemmas structures. Lemmas, in other words, wordnet representation of dictionary entry in canonical form. Lemmas has a built-in method of getting derivational forms, for which we searched for synsets to use in our calculation. If we found a verb or noun synset, we used it for similarity. In a case where we found both verb and noun, we would have used the one with higher score, but such a situation never arouses, but instead the derivative search only ever returned one word to use. Similarity score using wu-palmer had an average for all synonym groups of 0,715 with minimum and maximums of 0,125 and 1 respectively. Lin similarity average was 0,646 with minimum and maximums of 0 and 1. Word2vec method average was 0,363 and minimum/maximum being 0 and 1,011. Standard deviation for wu, lin and word2vec was in order 0,240/0,0303/0,315. Looking at the results, it should be pointed that this method most likely has little merit as incorporated here, as very often the derivative words of word-synonym pairs are the one and the same word, so similarity is naturally perfect. To avoid this, we would have had to look for more derived words but felt that then we would be just searching for more synonyms instead of comparing the once given which felt a bit beside the point.

E) Popularity correlation to similarity

This part explores the correlation between word similarity and their popularity. A natural starting point here is to first find the popularity of each word. We used the brown corpus, collection of American-english text samples, to get an estimate on popularity. Basically, we counted the frequency of the words in our study inside the corpus. This was bit intensive process to repeatetly count words from a collection of millions, so results were saved to python pickle object used to storing pythonic data models. We used the previously calculated wordnet similarities for similarity values, and then calculated their pearson correlation against popularity (Atoum & Otoom, 2016). Every synonym gets calculated absolute popularity distance to main word and then the synonym list is sorted based on it from largest difference to smallest. Thus, if there were correlation, highest similarity values would be at the end of list. Popularity scores were all over the place, from hundreads of hits to many words that never appeared in brown corpus. We did get the correlation values even so. Average pearson correlation for whole synonym group list is -0,023 with maximum of 1 and minimum of -1. There were some zeroes from synonym groups that had only one comparison word, but they were small enough group that it should not skew the comparison much. In any case, no correlation could be found.

F) Antonym similarity

Moving on from synonyms, we will next test if antonyms, words opposite in meaning to one another, have similar structure as words using the same methods as on our synonym investigation. Pulling the list of popular antonyms [10] was done by hand and they were formatted in a similar way to synonyms so that same scripting could be used to read the word list. To assess the similarity, Wu-Palmer via the wordnet was used first. Only nouns and verbs were used as the derivative search did not bring very promising results. We had 355 antonym pairs and wu-palmer was able to calculate results for 181. Average score was 0,424 with minimum of 0,100 and maximum of 0,900. Next, we did cosine distance calculation using word2vec. Methodology stayed the same as before. Average for word2vec was 1,008 with minimum of 0,733 and maximum of 1,351. Standard deviation was for wu-palmer 0,254 and word2vec 0,100. In addition to these calculations from previous tasks, we also calculated skew and kurtosis values for both similarity methods. Skew for wu-pal,er was 0,511 and for word2vec 0,243 and kurtosis was -1.122 for wu and -0,122 for word2vec. The results were very similar to our previous ones, meaning little similarity.

G) Antonym popularity

Continuing the ongoing trend, we now calculate popularity to our antonyms and check for correlation to similarity results. We again mined popularity values from brown corpus, ordered the word pairs based on how close they are in popularity and calculated their Pearson correlation values. The results we got were that Pearson correlation with wu-palmer similarity and popularity was -0,078 and with word2vec cosine distance the correlation was 0,002 meaning very weak correlation in both cases.

H) Glove embedding

We will also test antonym similarity correlation with popularity using glove embedding. GloVe or Global Vectors for word representation is learning algorithm for obtaining vector representation for words. We used pre trained model, trained with common crawl words, meaning data set of 42B tokens and 1,9M vocab. Algorithm is interesting but, in the end, we receive vectors similar to our word2vec method and we then calculate cosine distance with them like before. Similarity values using glove gave a minimum of 0,070, maximum of 0,846 and average of 0,414. Very different overall result to word2vec, most likely resulting for much larger training size for model. Pearson correlation for glove similarity and popularity was -0,063 so no correlation here either.

I) Duos vocabulary

Finally, we will repeat the process used for antonyms but this time we will use duos vocabulary [12]. Duos are pairs of words that often appear in collocation or together in the same order. For example, “Black and white” or “new and old”. Since they appear in pairs, we can basically treat them the exact same as the antonyms. Our list had 103 pairs of duos and 70 of them could be used in wu-palmer similarity considering the adjective problem. The results for wu palmer were as follows. Average of 0,485 and minimum/maximum values of 0,105/0,9 respectively. Standard deviation was 0,277. For word2vec cosine distance similarity the results were average of 1,001 with 0,803 minimum and 1,266 maximum. Standard deviation of 0,087. The results do not differ significantly from previous. Skew for Wu-Palmer is 0,172 and for word2vec it was 0,178. Kurtosis for Wu-Palmer is -1,435 and for word2vec 0,070.

Popularity results for duo pairs are as follows. Wordnet Wu-Palmer had Pearson correlation of -0,119. Of some note is that many words did not appear in brown dictionary at all, or they had very low popularity. This has been an issue in other word sets as well but here, 22 words from the start had no appearances and many pairs had appearances of only one of the words. For word2vec, Pearson correlation is -0,106. Low correlation for both similarity methods.

3 Results and discussion

3.1 Wu-Palmer similarity and Word2Vec

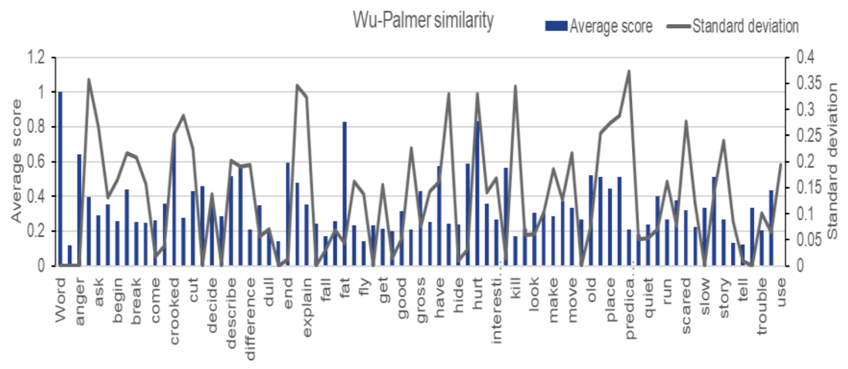


Figure 2. Average score and standard deviation for Wu-Palmer similarity

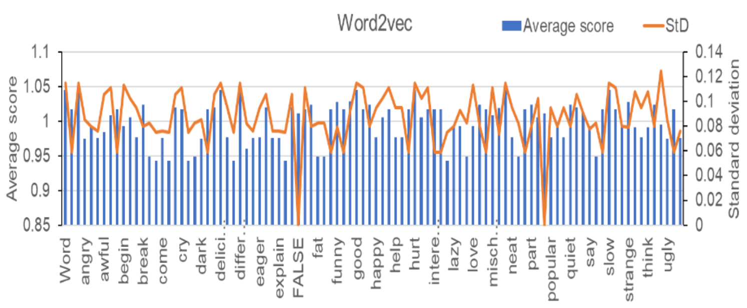
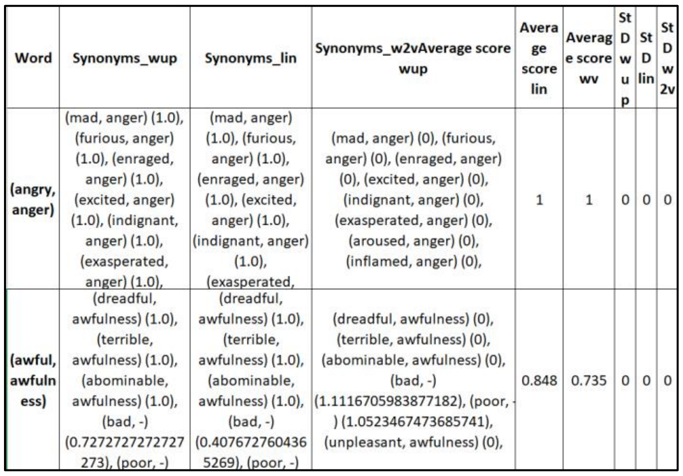


Figure 3. Average score and standard deviation for Word2Vec similarity



Word	Synonyms_wup	Synonyms_lin	Synonyms_w2vAverage score wup	Avera ge score lin	Averag e score wv	St D w u p	St D lin	St D w 2v
(angry, anger)	(mad, anger) (1.0), (furious, anger) (1.0), (enraged, anger) (1.0), (excited, anger) (1.0), (indignant, anger) (1.0), (exasperated, anger) (1.0),	(mad, anger) (1.0), (furious, anger) (1.0), (enraged, anger) (1.0), (excited, anger) (1.0), (indignant, anger) (1.0), (exasperated,	(mad, anger) (0), (furious, anger) (0), (enraged, anger) (0), (excited, anger) (0), (indignant, anger) (0), (exasperated, anger) (0), (aroused, anger) (0), (inflamed, anger) (0),	1	1	0	0	0
(awful, awfulness)	(dreadful, awfulness) (1.0), (terrible, awfulness) (1.0), (abominable, awfulness) (1.0), (bad, -) (0.7272727272727273), (poor, -)	(dreadful, awfulness) (1.0), (terrible, awfulness) (1.0), (abominable, awfulness) (1.0), (bad, -) (0.4076727604365269), (poor, -)	(dreadful, awfulness) (0), (terrible, awfulness) (0), (abominable, awfulness) (0), (bad, -) (1.1116705983877182), (poor, -) (1.0523467473685741), (unpleasant, awfulness) (0),	0.848	0.735	0	0	0

Table 2 sample results of synonyms and its average score (Full results are given as annexure)

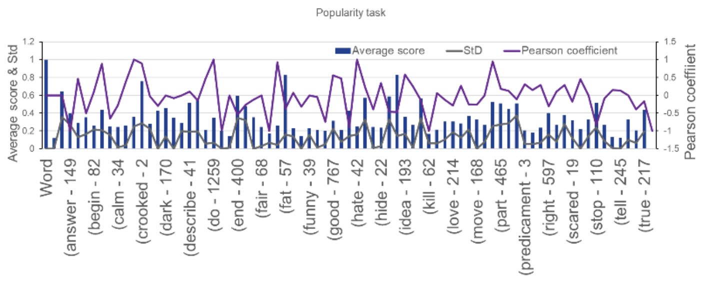


Figure 4. Popularity of the words and its average score and standard deviation

3.2 Discussion

The results for wordnet and Wu/Palmer similarity were rather low. If exact similarity would result in score of 1, most of our results were on average below 0,5. Even more troubling, our scores were very distributed basically all over the place, whereas if there were consistent similarity, they would be more grouped. That said, there were many words with high scores as well, so some similarity existed, but too little to say that it was not coincidental. Word2vec gave more consistent results, but somewhat worryingly, the averages were around 1 in all groups. This might indicate error in calculation, but further investigation might be needed. Even better prove of error, glove calculation gave average for antonyms of 0,4 even though it too was vector distance between words. Correlations between popularity and similarity were very low across the board, independent of similarity calculation method used. Going forward, using the pre/trained glove vector algorithm would be prudent for more trustworthy results. Doing analysis over antonyms and duos seemed to be somewhat separate matter from our goal of synonym research, but the results could admittedly have proven interesting similarities or correlations, so it was good that those were checked.

For representing the results, some kind of interactive gui could certainly end up as useful. In ideal case, we would have gui that have separate pages for synonyms, antonyms and duos. Pages would have result tables where each word would have their related statistics like popularity and similarity scores to their corresponding word in all the used methods. Some kind of summary page showing scores from all datasets would also be rather beneficial. If we really want to take this to the end, interactive word search and live similarity calculation for given words would be easy enough to add on top, even if it would go a bit beyond the scope of this project.

For further investigation, larger list of synonyms would be beneficial, as 96 most popular might not be enough to say conclusively one way or another, even if it is not looking too good at the moment. Same increase of datasets would probably benefit antonyms and duos as well. And since we are already studying three separate word groups, investigating if there are additional word pair groups and studying those might prove interesting.

4 Conclusion

Semantic Similarity provides a useful tool in identifying the closeness of similar words and texts and thus is very important for information retrieval. This project has conducted different methods of retrieving semantic similarity measurement of common English words. The datasets used were rather small in sizes. Size of datasets was commonly only few hundreds of word groups. Even so, if there were strong similarities in these groups, we most likely would have seen a stronger showing of it. As it were, there likely is little to none expected similarity although further testing with larger datasets could be beneficial.

References

1. What is WordNet? https://wordnet.princeton.edu/
2. Christopher D. Manning & Hinrich Schutze, Foundations of statistical natural language processing, The MIT Press, Cambridge, London, England
3. Lingling Meng, Runqing Huang and Junzhong Gu, A Review of Semantic Similarity Measures in WordNet, International Journal of Hybrid Information Technology Vol. 6, No. 1, January, 2013
4. Aboelela E.M., Gad W., Ismail R. (2020) Feature Extraction Using Semantic Similarity. In: Hassanien A., Shaalan K., Tolba M. (eds) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019. AISI 2019. Advances in Intelligent Systems and Computing, vol 1058. Springer, Cham. https://doi.org/10.1007/978-3-030-31129-2_8
5. Montserrat Batet & David Sánchez, A review on semantic similarity, IGI Global Microsoft Word 2007 Template
6. Pawan Sharma, Rashmi Tripathi, Vivek K. Singh, R.C.Tripathi, ‘Automated Patents Search through Semantic Similarity’, IEEE International Conference on Computer, Communication and Control (IC4-2015)
7. Ali Sebti, Ahmad Abodollahzadeh Barfroush, A new word Sense similarity measure in WordNet, Proceeding of International Multiconference on Computer Science and Information Technology, pp, 369-373
8. Issa Atoum, Ahmed Otoom, Efficient Hybrid Semantic Text Similarity using Wordnet and a Corpus, *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 9, 2016*
9. Wayan Simri Wicaksana and Bambang Wahyudi, ‘Comparison Latent Semantic and WordNet Approach for Semantic Similarity Calculation’, *Gunadarma University Jl. Margonda Raya 100, Depok, Indonesia*

10. https://www.enchantedlearning.com/wordlist/opposites.shtml Common Opposites - Antonyms Vocabulary Word List
11. https://justenglish.me/2014/04/18/synonyms-for-the-96-most-commonly-used-words-in-english/ Synonyms for the 96 most commonly used words in English
12. https://www.enchantedlearning.com/wordlist/duos.shtml Famous and Common Duos Vocabulary Word List