ANLP Assignment 2: Word Similarity Measures

Antonella Donvito, Ricardo Muñoz Sánchez

1 Introduction

Different methods can be employed in NLP to investigate similarity between words, defining two words similar if they occur in similar contexts. We must first find a vector representation of the contexts in which a word appears, and then compare such vectors. In our investigation, we decided to keep fixed the representation of the word embedding vectors, using Positive Pointwise Mutual Information (PPMI) as a measure of dependency between words, as it is one the best metrics in NLP literature [**1**]. On the other hand, we wanted to investigate how different similarity metrics between the sparse embedded vectors affect the computation of similarity between words. To answer this question, we compared cosine similarity, weighted Jaccard similarity and Pearson correlation. Furthermore, we wanted to explore whether clusters would form when using dimensionality reduction. For this, we used the Uniform Manifold Approximation and Projection (UMAP) dimensionality reduction technique on the PPMI vectors.

2 Methodology

To select our test words, we first chose six semantic categories. Then, for each category we chose 10 words so that there are words with different frequency rates and some that we would expect to see in more than one category. In this last case, we added the word to the category we thought would fit the best. The list of words and the categories we fit them in can be found in Table 1.

In order to compare how similar these words are, we first obtained the PPMI vectors from the co-ocurrence vectors. We then tried three ways to check the proximity of two vectors, namely cosine similarity, Jaccard similarity and Pearson's correlation.

Jaccard similarity measures the size of the intersection between two sets. The usual version just takes into account whether there is an element in the intersection, while the weighted version (also called Ruzicka similarity [**3**]) also takes into account the number of co-occurrences and not just whether they appeared together. It is given by $J_{\mathcal{W}} = (\sum_i \min(x_i, y_i))/(\sum_i \max(x_i, y_i))$ [**4**]. We had previously considered that two words are similar if the context that they appear on is similar and we believe that just checking how large are these intersections is a very intuitive way to do that.

Pearson's correlation coefficient, on the other hand, measures whether there is a linear correlation between the two vectors. This is what we would expect from two similar PPMI vectors, as they should be pointing in similar directions, despite their length [**5**].

In order to check whether the words that we chose did in fact cluster together, we used two visualization methods. The first one was heatmaps, in which the diagonal was obscured so that the other values could be seen more clearly. The other one was using the dimensionality reduction algorithm UMAP. We also created word clouds for each similarity measure for a target word (we chose *pizza*) with the the 40 most similar words to it. This was to better illustrate visually the implications that these different measures have on altering the most similar words to another one.

3 Results

Table 4 shows the 10 highest similarity scores for each similarity measure. In general, the cosine similarity scores are higher than the others, with 0.50 being the highest. Although we cannot find any specific patterns in these results, all the pairs look significant, with the words actually being similar as expected.

Figure 1 shows one heatmap for each similarity measure. As expected, the cosine similarity heatmap is the brightest, while the Jaccard one is the darkest. More interestingly, the scores are higher around the main diagonal, suggesting higher similarities within the clusters, with some exceptions. On the other hand, the Jaccard similarity heatmap presents many bright spots outside the diagonal, which implies relativelty high similarities across clusters.

While the heatmaps do a good job in showing the general patterns of the data, we wanted to represent our PPMI sparse vectors into a 2-dimensional space. Since the traditional PCA turned out to be not efficient in reducing the feature space in a meaningful way, we used UMAP, a dimensionality reduction algorithm that aims to capture the local topology of the data. The full mathematical explanation of the algorithm is outside the scope of this essay, but further details can be found in the original paper [**2**]. By looking at the UMAP plots (Figure **??**), we can see that the words are mostly grouped in the same clusters as the ones that we assigned to each word. Not surprisingly, most of the exceptions are represented by the ambiguous words. An example is the word *cookie*, that we considered as a tech term, but was clustered with other food instances regardless of the distance metric. Overall, consistently with the previous results, the worst representation is obtained using Jaccard similarity, with the data points not being grouped into very distinct blocks.

Finally, Figure 2 shows the wordclouds for the word *pizza* that were generated using the similarity scores, with bigger words being more similar to the target word. We can see that most of the words in the clouds are indeed related to our target, with the most notable exception being the Jaccard similarity one, which a high number of unrelated words.

4 Conclusion

Through our exploratory analysis, we showed how different similarity metrics between vectors can affect the computed similarity between words. Since the absolute similarity score values may be hard to interpret, we chose to draw attention on semantic clusters and on data visualization techniques for evaluation. In this regard, cosine similarity and correlation show similar pattern. This can be explained by the fact that both metrics measure linear relations, while Jaccard similarity is a measure of intersectional cardinality. Correlation turned out to be a good metric also in terms of computational costs, compared to cosine similarity.

An interesting way to expand our work would be using vector similarity and distance metrics available in literature [**3**], alongside with substituting PPMI with alternative word-context metrics. Also, we illustrated that these metrics are better at capturing context similarity rather than the actual word similarity. To give a couple of examples, *Mubarak*, a personal name, and *Tahrir*, the name of a place, are rated as similar and are clustered together, while the synonymity relation between *mouse* and *rat* was not really captured. Hence, we believe that it is of critical importance to be conscious of what these similarity metrics are measuring, as the concept of similarity we used might turn out to be different than our intuitive idea of similarity as humans.

Appendix A - Figures and Tables

BIEBER	Belieber, baby, selena, #myworldtour, sing
	love, hate, beautiful, stratford
#arabspring	#wallstreet, revolution, #occupy, mubarak
	twitter, muslim, tahrir, revolucion, Egypt
TECHNOLOGY	computer, iphone, android, apple, ford
	car, mustang, cookie, mouse
ANIMALS	dog, cat, pet, turkey, bug
	rail, chicken, wombat, rat
FOOD	eat, coffee, quinoa, stock, pizza
	good, goood, goooood, macoroni
FAMILY	mom, dad, kids, offspring, divorce
	wife, husband, cuute, matron

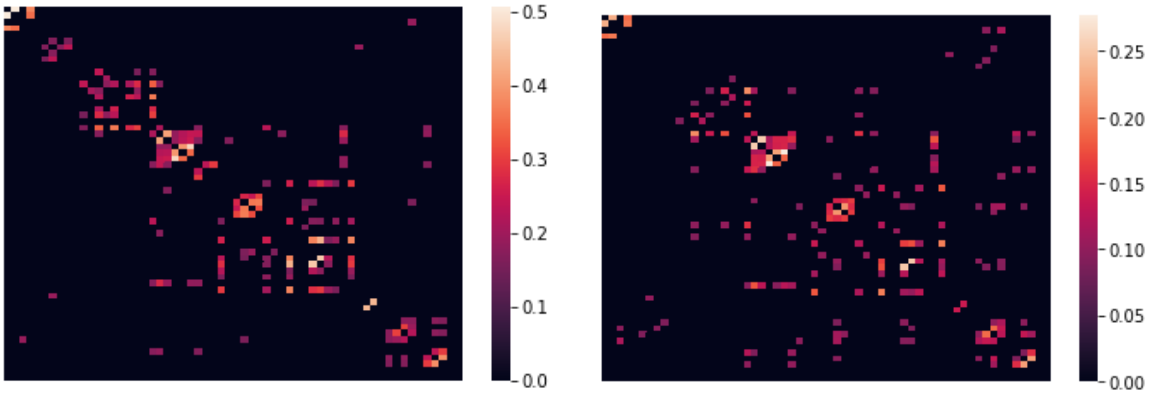
Table 1: Test words and categories.

Pair	Cosine Sim.
bieber, belieber	0.505
iphone, apple	0.479
food, eat	0.459
goooood, goood	0.437
eat, chicken	0.430
computer, technology	0.410
iphone, android	0.398
husband, wife	0.388
bieber, selena	0.378
pizza, chicken	0.376

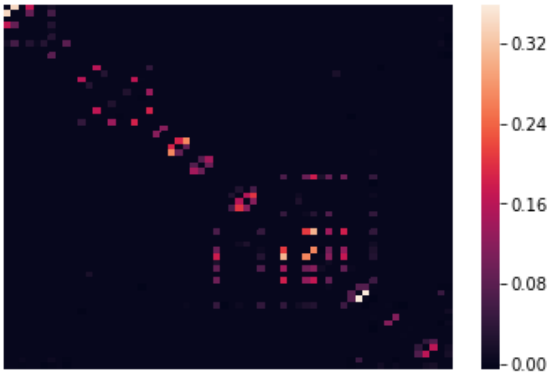
Pair	Jaccard Sim.
iphone, apple	0.276
computer, technology	0.254
food, eat	0.251
bieber, belieber	0.233
iphone, android	0.224
husband, wife	0.223
cat, dog	0.217
revolution, egypt	0.211
pizza, chicken	0.203
believe, selena	0.199

Pair	Correlation
goooood goood	0.359
bieber, belieber	0.341
eat, chicken	0.306
iphone, apple	0.268
food, eat	0.266
food, chicken	0.208
dog, pet	0.204
iphone, android	0.187
egypt, tahrir	0.18
pizza, chicken	0.181

Table 2: Top 10 similarities for each measure

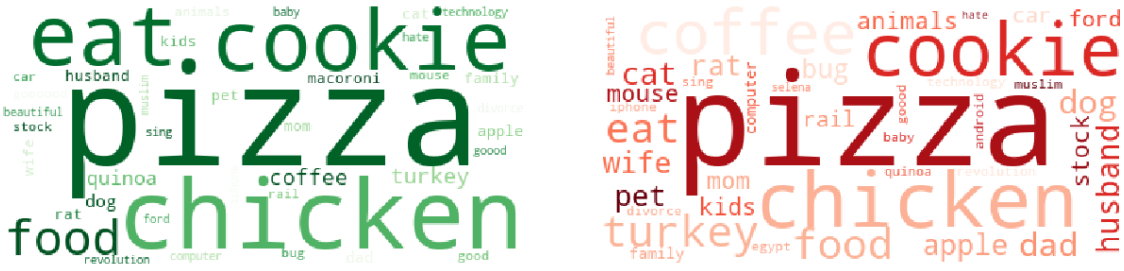


Cosine Similarity Jaccard Similarity



Correlation

Figure 1: Similarity Heatmaps



Cosine Similarity Jaccard Similarity

Correlation

Figure 2: The wordclouds for *pizza* with the different measures

Pair	Similarity
cat, dog	0.361
computer, mouse	0.167
cat, mouse	0.121
mouse, dog	0.0909
cat, computer	0.072
computer, dog	0.061
@justinbieber, dog	0.0155
cat, @justinbieber	0.015
@justinbieber, computer	0.014
@justinbieber, mouse	0.009

Table 3: Preliminary task similarity results.

Appendix B - Dimensionality reduction with UMAP

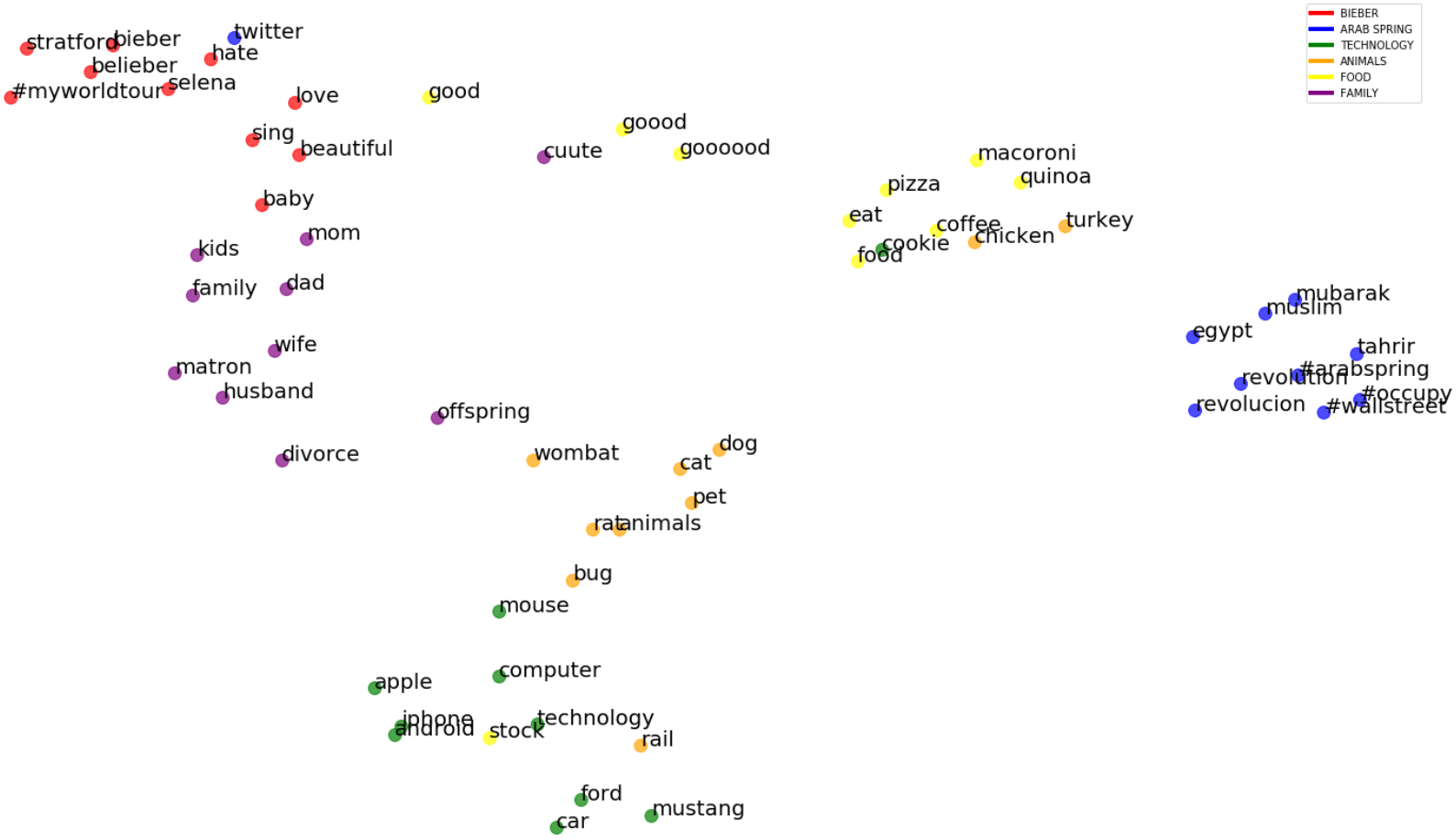


Figure 3: Cosine Similarity

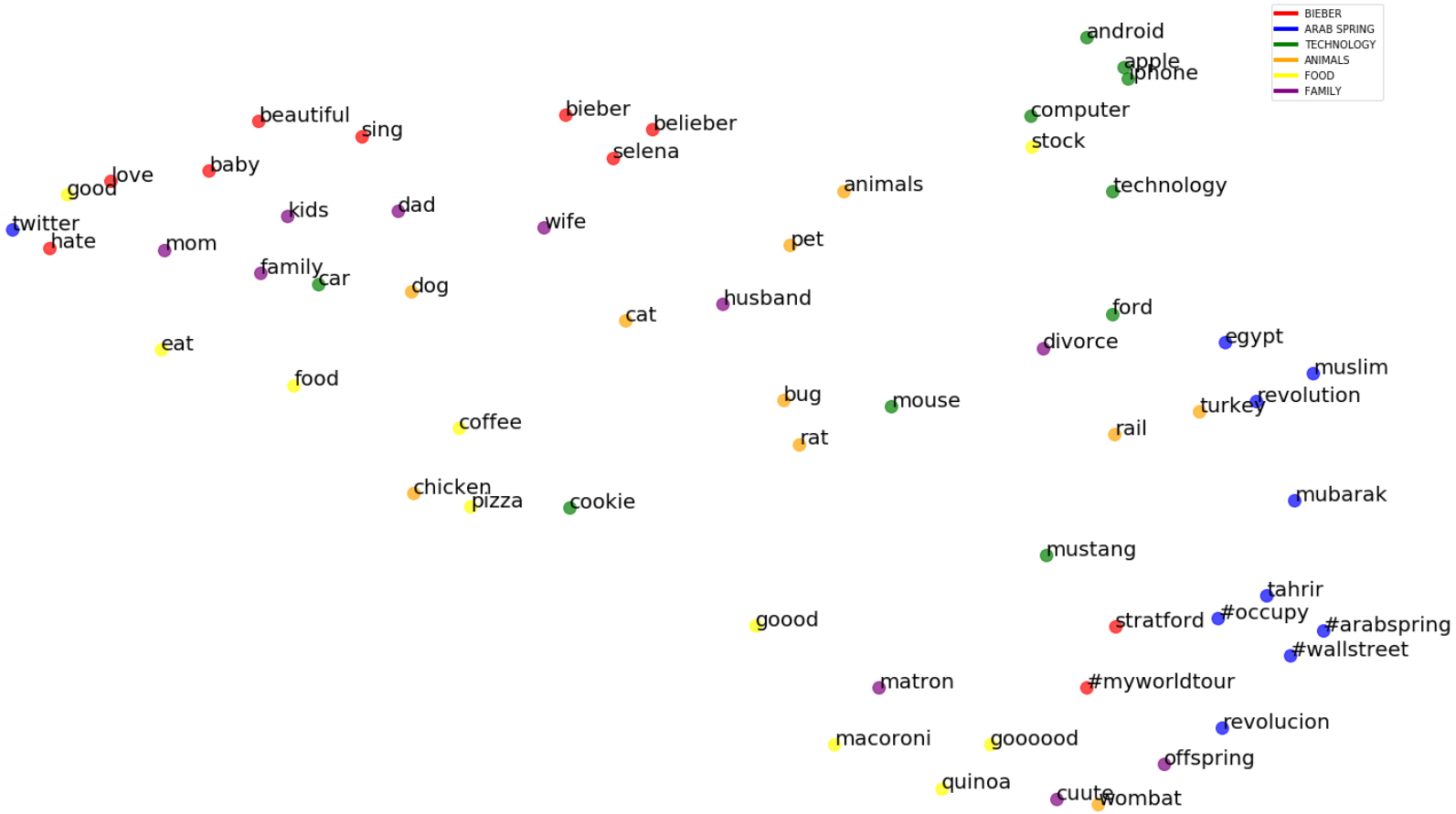


Figure 4: Jaccard Similarity

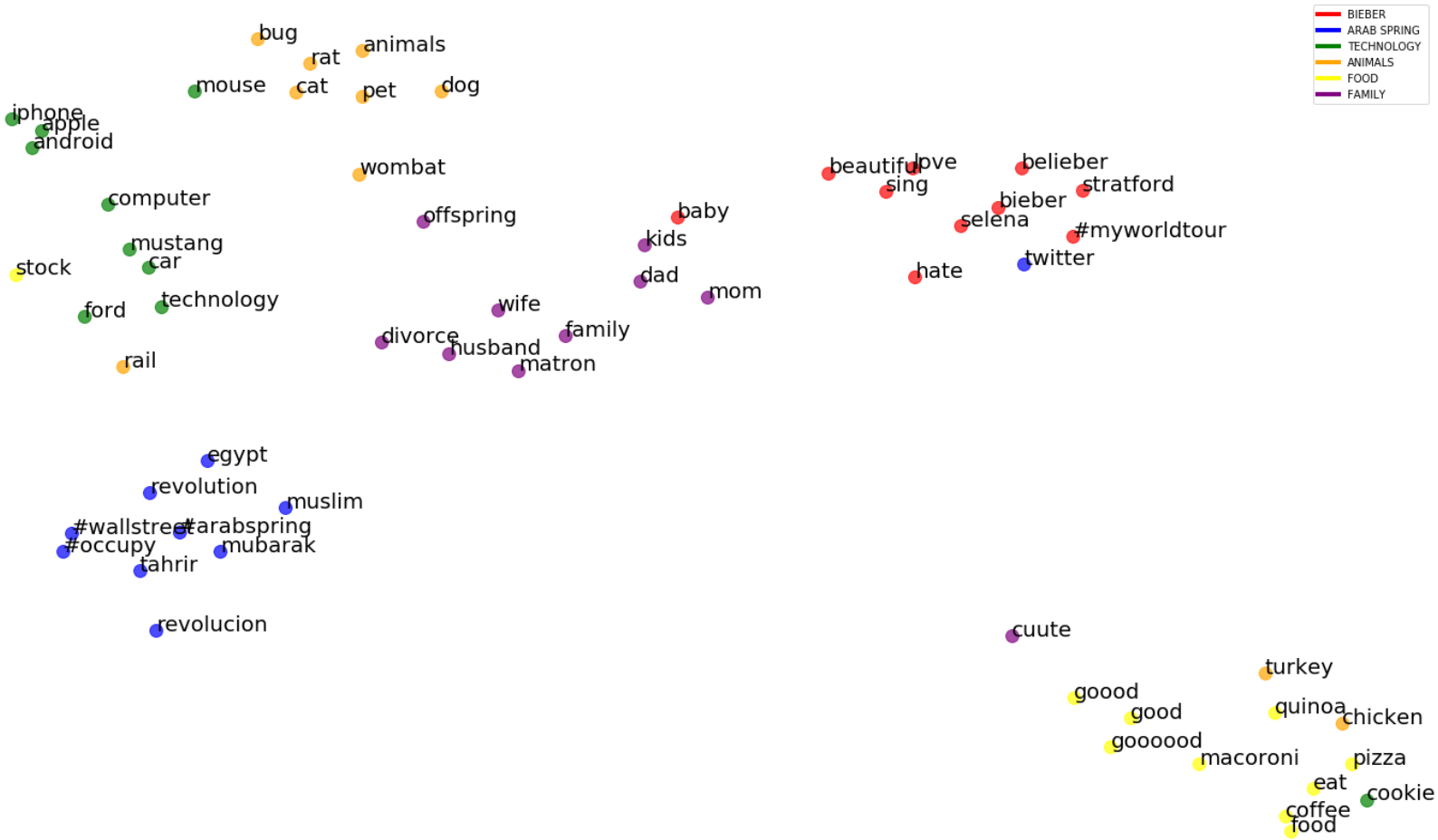


Figure 5: Correlation

References

[1] Bullinaria, J.A. & Levy, J.P. (2007). Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study. Behavior Research Methods, 39, 510-526.

[2] McInnes, L., Healy, J. & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.

[3] Cha, Sung-Hyuk (2007). Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions.

[4] Ioffe, Sergey (2010). Improved Consistent Sampling, Weighted Minhash and L1 Sketching. 2010 IEEE International Conference on Data Mining, 246-255.

[5] Peter D. Turney and Patrick Pantel (2010). From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research, 37(1):141–188

[6] Jurafsky, Dan (2015). Computational Lexical Semantic, Lecture 3: Vector Semantics [PowerPoint Slides]. Retrieved from https://web.stanford.edu/∼jurafsky/li15/lec3.vector.pdf on 2019/11/25

[7] Jurafsky, M., Martin, J.H. (2019). Speech and Language Processing, 3rd ed. draft.

[8] Egghe, L., Leydesdorff, L. (2009). The relation between Pearson's correlation coefficient r and Salton's cosine measure.