Measuring the Effectiveness of Word-Embeddings for Facet Completion Tasks

Master thesis by
Prem Kumar Tiwari

December 1, 2020

First Referee: Prof. Dr. Benno Stein
Second Referee: Prof. Dr. Andreas Jakoby

Supervisor: Tim Gollub

Bauhaus-Universität Weimar

Table of Contents

01 Motivation

02 Challenges and Related Tasks

03 Dataset Preparation from WordNet

04 Effectiveness Evaluation

05 Developing Embeddings Model for Facet Generation Tasks

06 Conclusion

01 Motivation

What is Facet Term Recommendation?

It is a process of suggesting entities which belong to same semantic concept. The relevance of facets depends **upon the information needed by the users**.

User Query: Erfurt **Recommendation 1**: Weimar, Jena. . . **Recommendation 2**: Berlin, Magdeburg, Munich. . . **Recommendation 3**: **E**astbourne, **E**ssen, **E**tah. . .	**User Query**: Erfurt, Weimar **Recommendation**: Eisenach, Jena. . .

Why Do we need Facets?

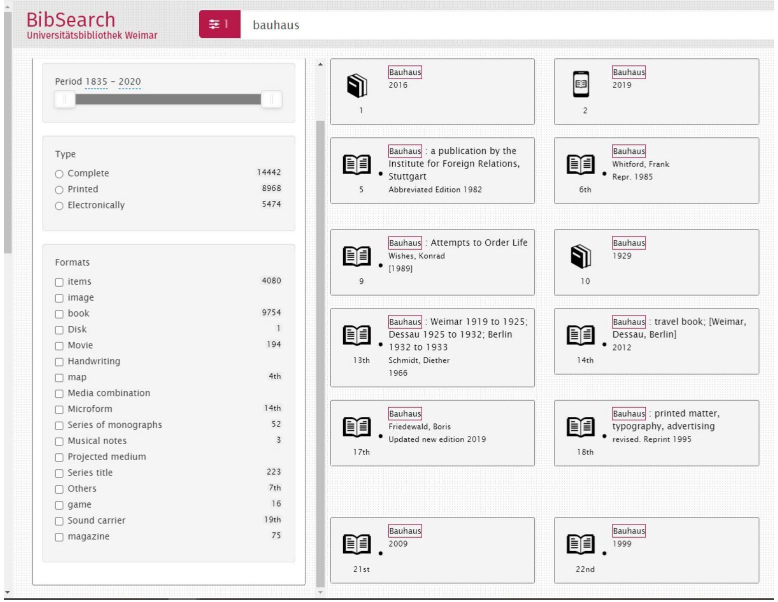
Motivation

Use Cases:

1. Facets are used in Faceted Search System

https://bibsearch.uni-weimar.de/bauhaus/q=q1&fy=183518352020 2020&fb=10002020&fd=local&fp=0&fs=0&ff=0

But users want different facets, e.g. Cities.



Feddoul, L. et al. [1] points out that manual predefinition is often inappropriate and, apt choosing among facets is **virtually impossible** without **algorithmic support**.

A pre-definition of all possible Facets is impossible, because users have many different and individual needs. Hence, the need for **Algorithmic Support**.

Existing Algorithmic support for Facet Term recommendation Tasks

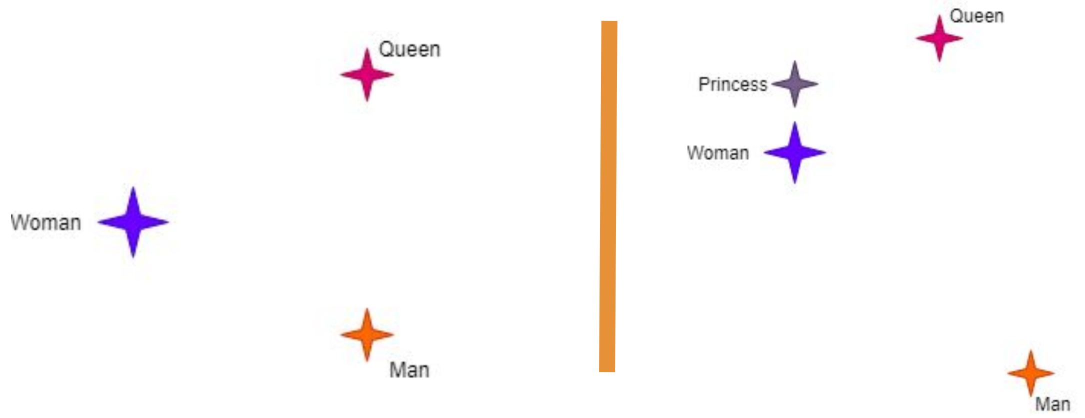
Faceted Browsing over Knowledge graphs[2] but they are predefined static structures, i.e. don't alleviate the problem of individual user needs.

So, how can we recommend?

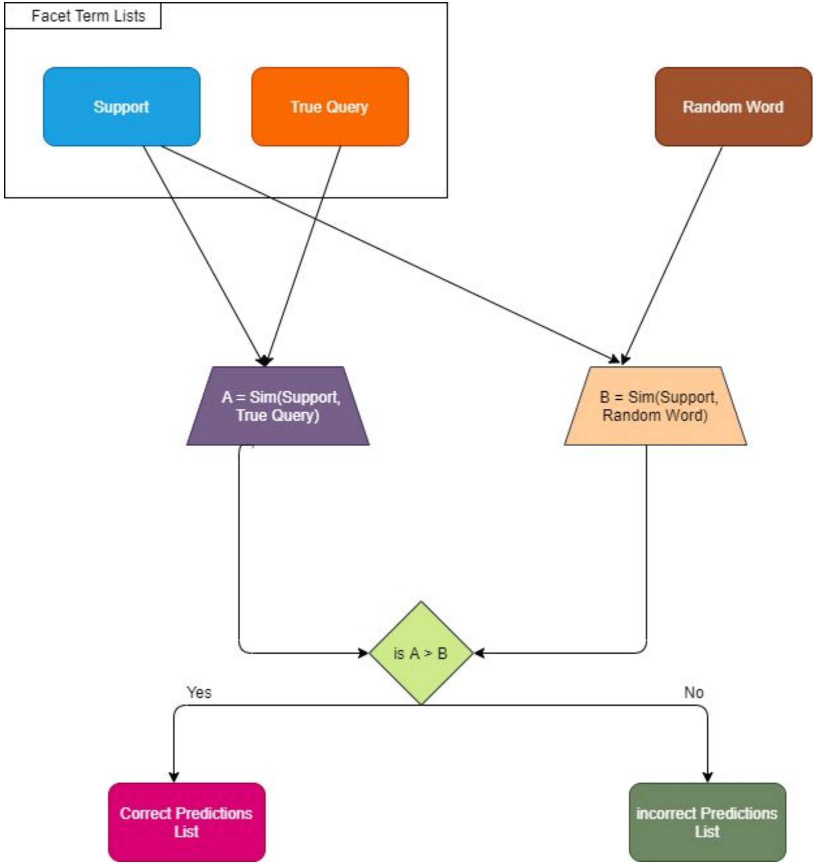
Why Word-Embeddings for Facet Term Recommendation Tasks?

Word- Embeddings algorithm **provide similar representation** in a vector space for words that are similar to each other.

Through this we can navigate the vector space when **word sense disambiguation** increases and hence use the algorithmic support of word-embeddings for the tasks of facet term recommendation tasks.



Now we would like to evaluate how word-embeddings performs on the task of Facet Term recommendations. For this we need a **procedure** and a **dataset**.



How we do it?

We assess the cosine similarity among facets Terms and pit it against cosine similarity among random words.

We further analyze the similarity distribution among these Facet terms to understand the efficiency of word-embeddings model for this tasks.

DataSet Preparation

For the purpose of understanding distribution of Facet Terms in the word-embeddings model, we shall create **Facet** from WordNet and then analyze the similarity distribution among the facet Terms of these Term Lists.

What is **Facet**?

It is one of the representation of knowledge structures. Term Lists(Facets) contains list of Facet Terms. We shall create Term Lists from another form of knowledge structures - Term Hierarchies- WordNet.

For the purpose of **creating Facets from Wordnet**, we shall only be working with 'Nouns' (POS).

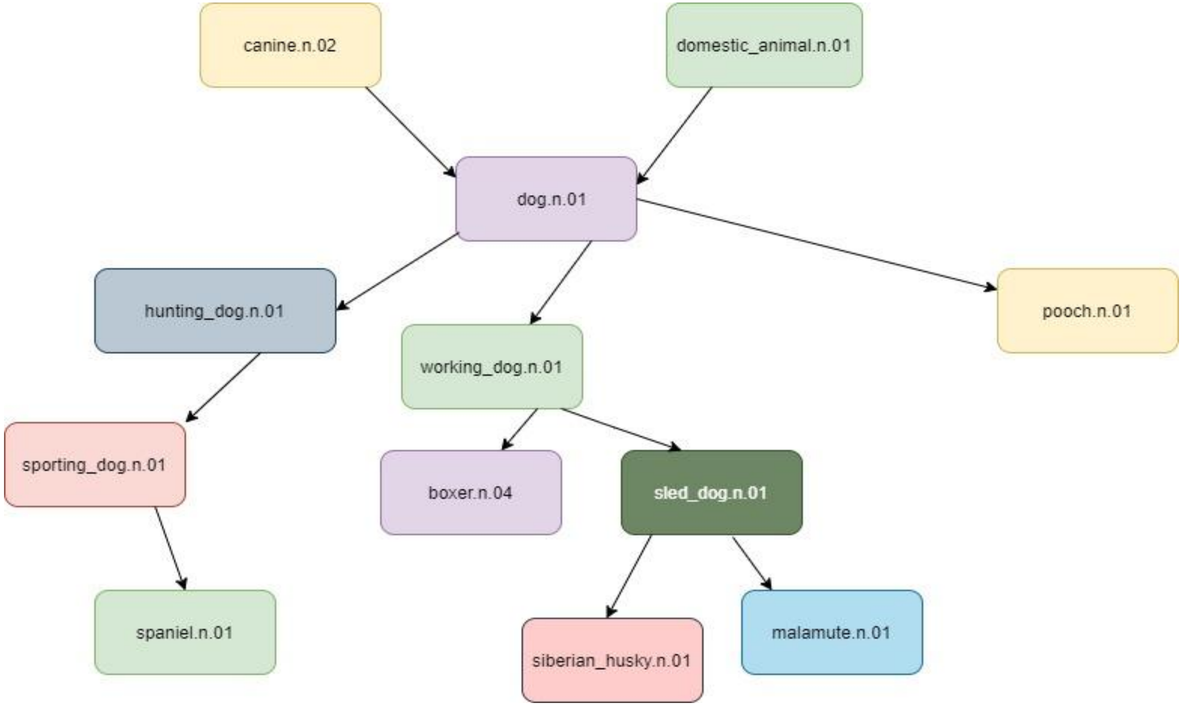
03 Dataset Preparation from WordNet

General Overview

This section is divided into two parts:

1. Extracting Facet Term Lists by Traversing WordNet.

2. From the Facet Term Lists, we generate Support-Query that can be fed to the evaluation procedure as shown in previous slide.

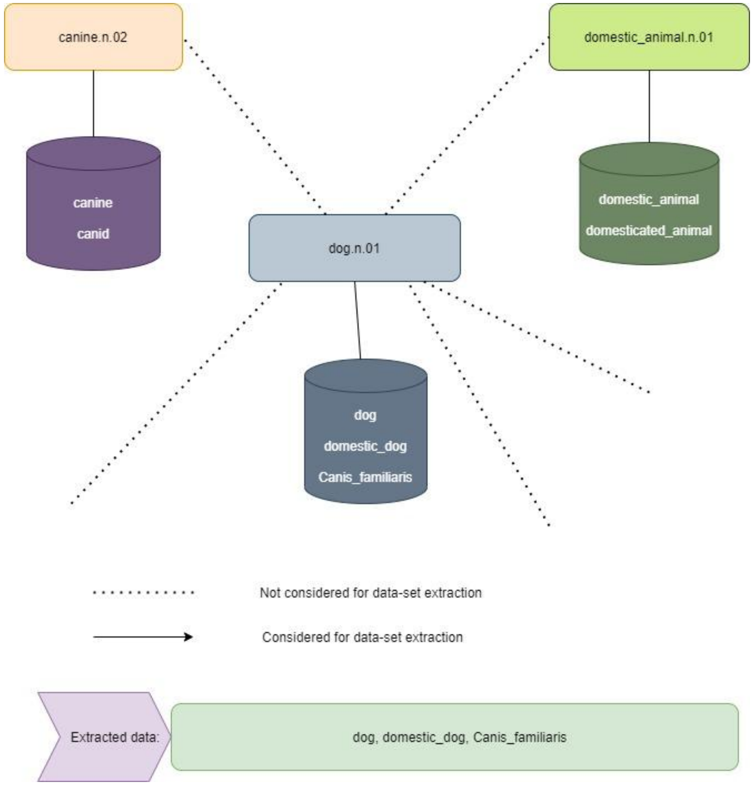


Possible ways of Term Lists Generation

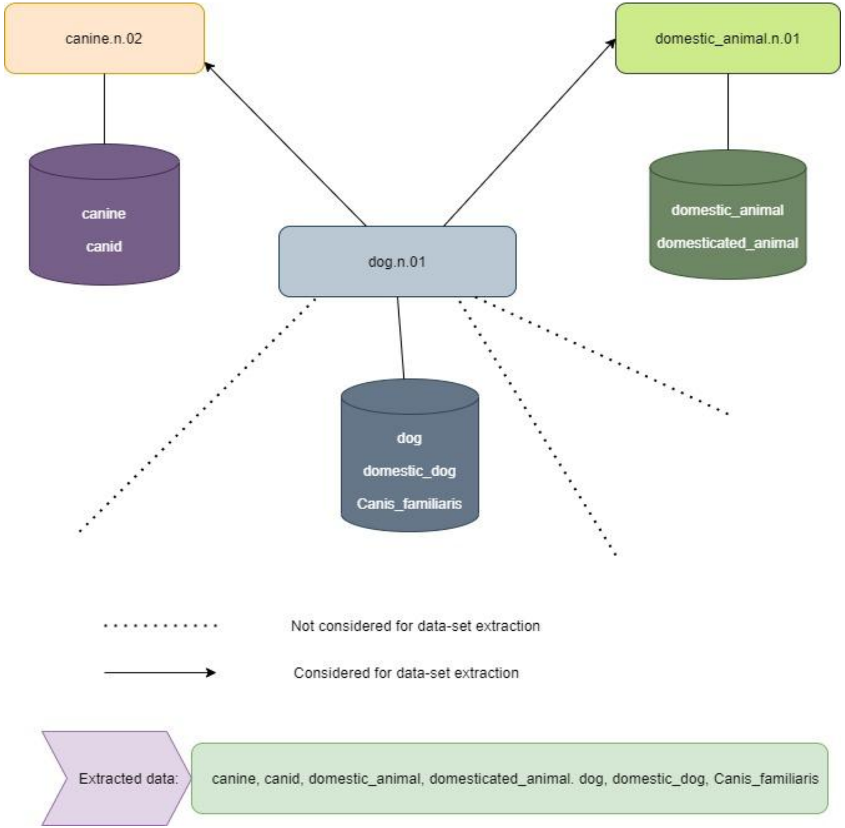
1. Direct Synset
2. Hypernym Traversal
3. Hyponym Traversal

Term Lists Preparation Directly from Synset and Hypernymy Traversal

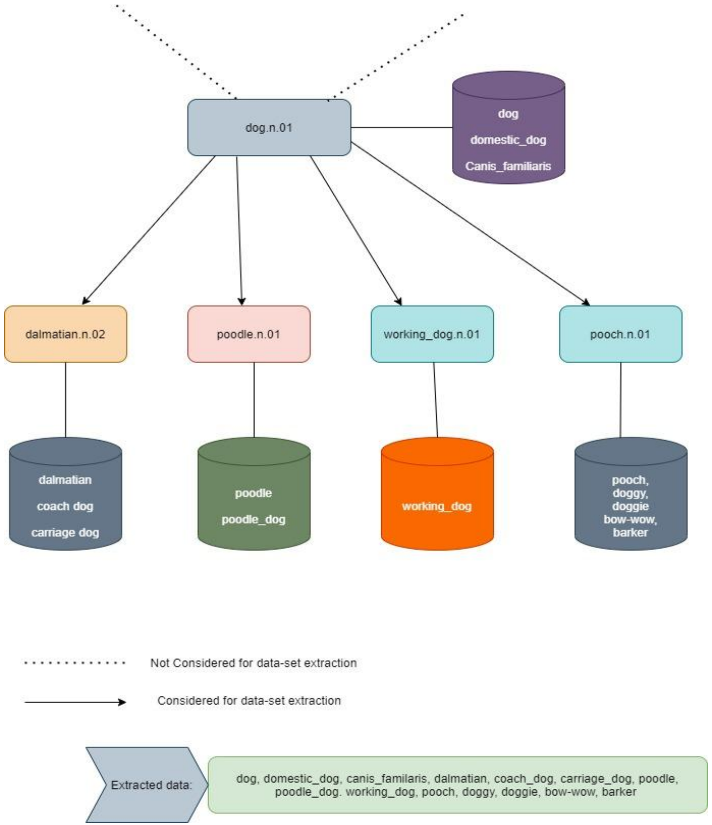
Direct Synset



Hypernymy Traversal



Term Lists Preparation through Hyponymy Traversal: Not Considered



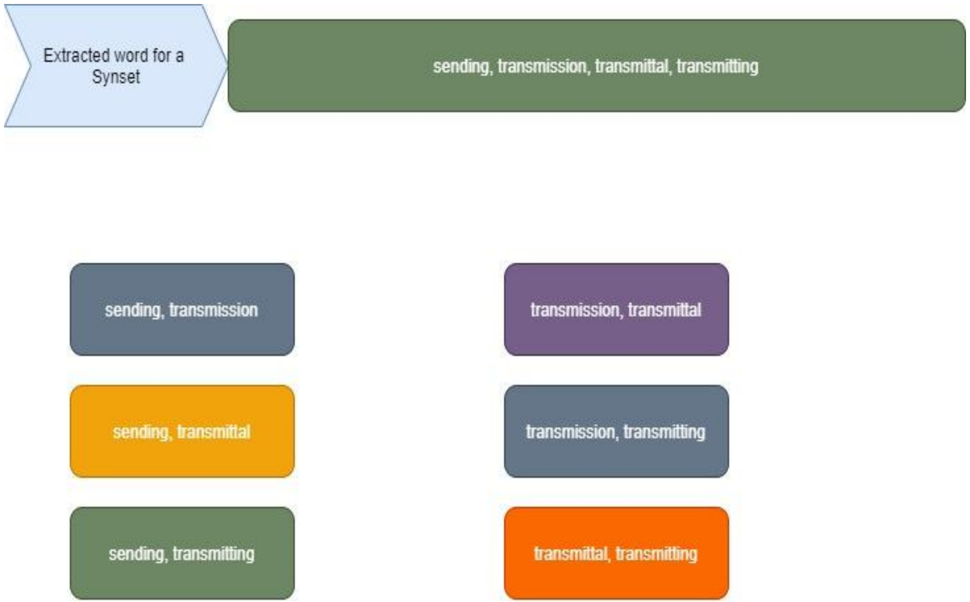
Support - Query Generation

From the generated Term Lists, we generate our Support-Query pair.

Two Types

1. Single Support(**X**) and Single True Query(**Y**).
2. Multiple Support(**X**) and Single True Query(**Y**).

Single Support(X) and Single True Query(Y).



C (4,2) = 6 i.e 6 combination pairs of two is generated from 4 words for a given synset.

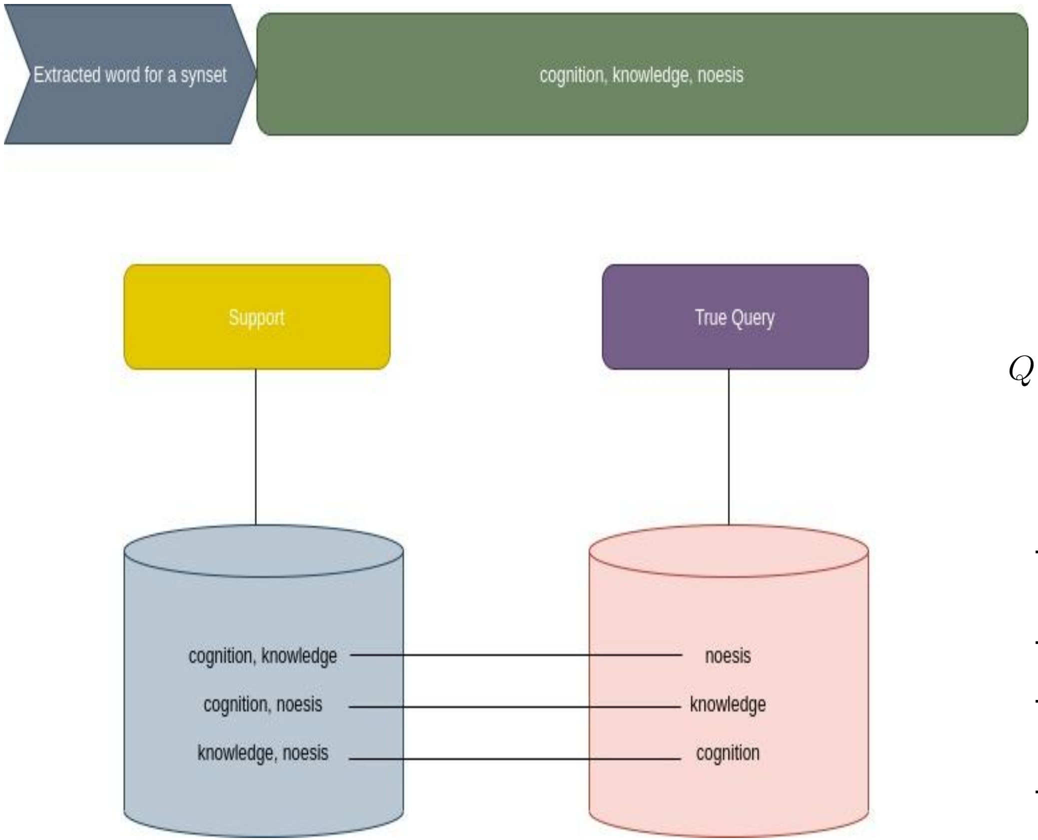
$$C(n, 2) = \frac{n!}{2!\,(n-2)!}$$

For every Term Lists generated, we generate **Support-Query** as shown in diagram.

$$C(n,2) = \frac{factorial(n)}{factorial(n-2) * factorial(2)}$$

Type of WordNet Traversal	Number of (Support-Query)Pairs generated
Preparation from Direct Synset	107469
Preparation from Hypernymy traversal of WordNet	366712

Multiple Support(X) and Single True Query(Y).

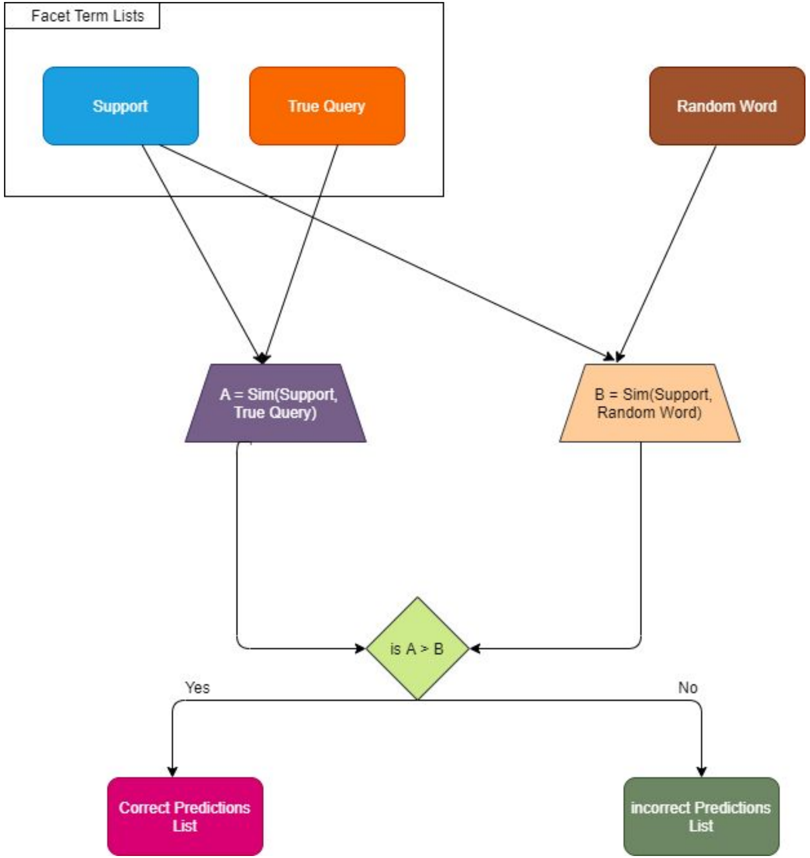


For every Term Lists generated, we generate **Support-Query** as shown in diagram.

$$Q(n,2) = (n-2) * \frac{factorial(n)}{factorial(n-2) * factorial(2)}$$

Type of WordNet Traversal	Number of (Multiple Support-Single Query) generated
Preparation from Direct Synset	258408
Preparation from Hypernymy traversal of WordNet	1380524

04 Effectiveness Evaluation

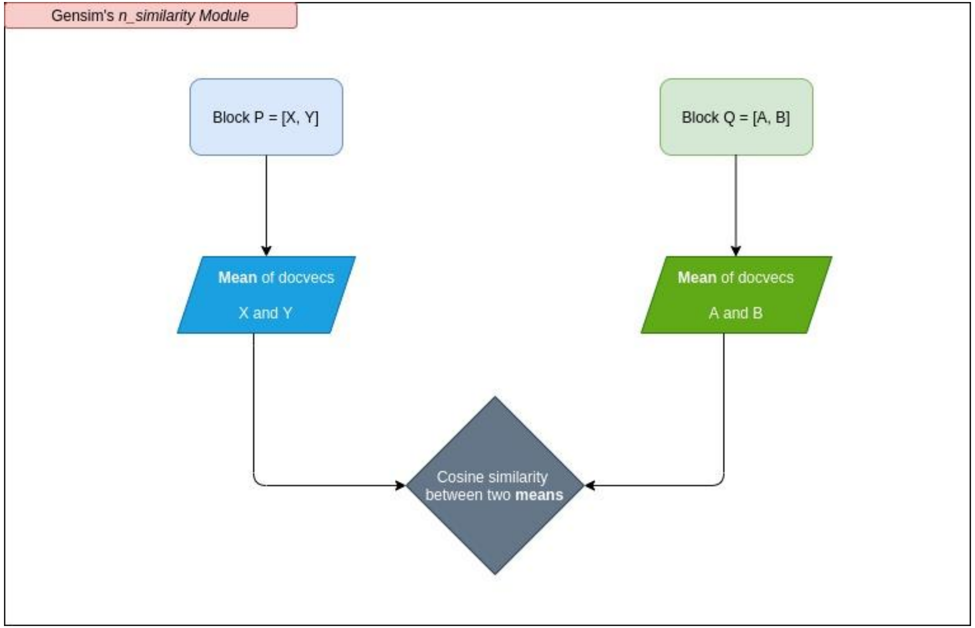


Two Types

1. Single Support(**X**) and Single True Query(**Y**).
2. Multiple Support(**X**) and Single True Query(**Y**).

We make comparisons for all the dataset prepared as shown in previous two pages according to procedure shown here.

How is Similarity Calculated?



Block P gets Support (Single or Multiple)

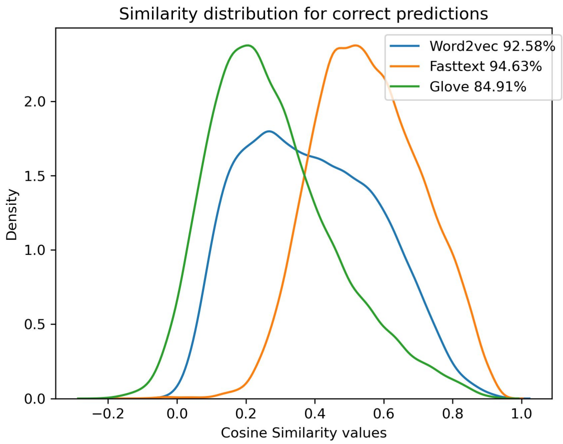
Block Q gets either True Query or Random word

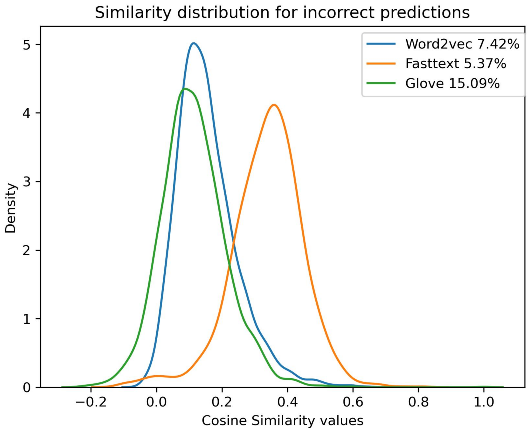
After running the procedure, we take the similarity values and plot them for both correct and incorrect predictions. Their respective accuracy on the dataset prepared from WordNet is also shown in the figures that follow.

Single Support(**X**) and Single True Query(**Y**) for Direct Synset.

Direct Synset

RESULTS ANALYSIS

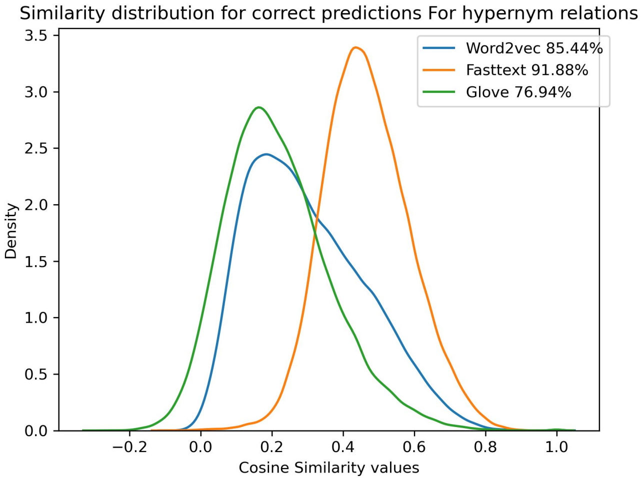


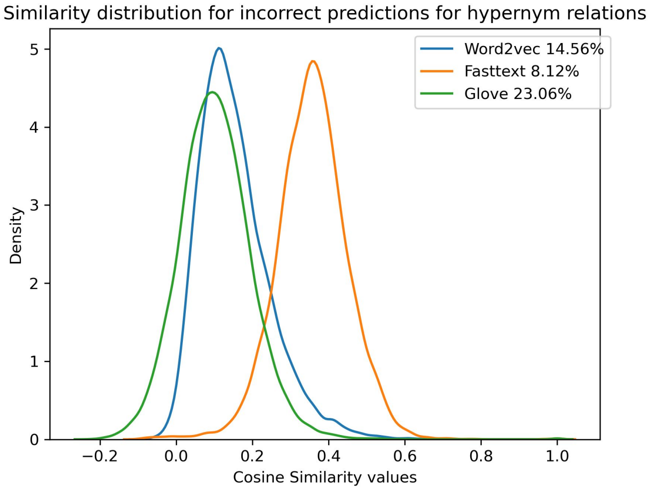


RESULTS ANALYSIS

Hypernym Traversal

Single Support(**X**) and Single True Query(**Y**) for Hypernym Traversal.

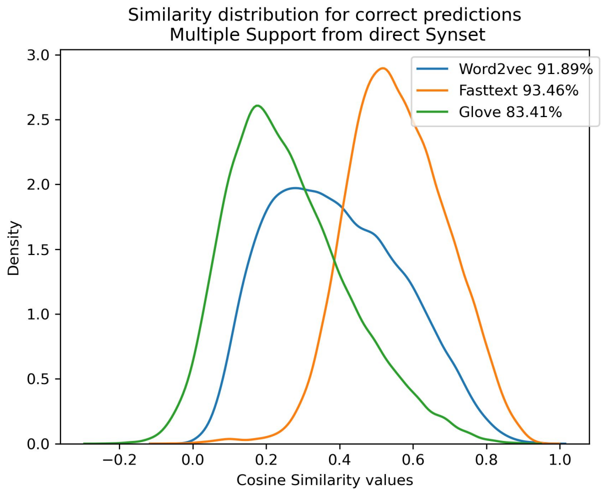


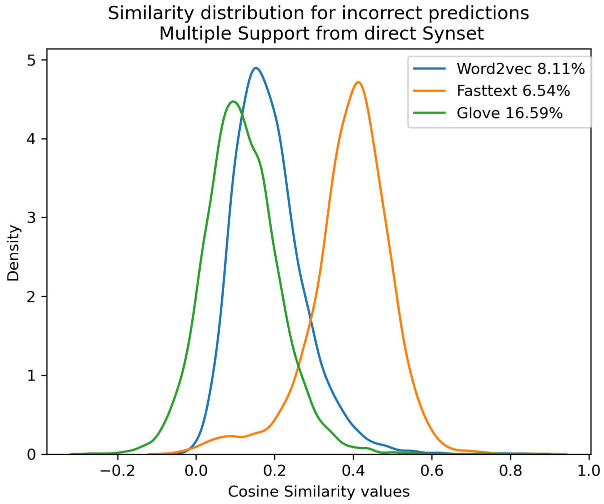


RESULTS ANALYSIS

Direct Synset

Multiple Support(**X**) and Single True Query(**Y**) for Direct Synset.

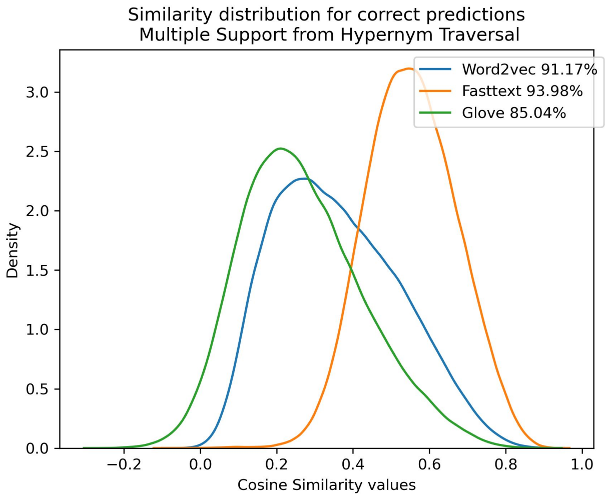


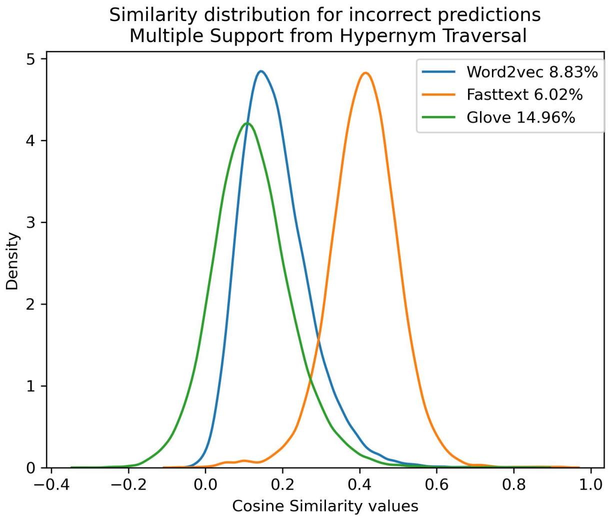


Hypernym Traversal

RESULTS ANALYSIS

Multiple Support(**X**) and Single True Query(**Y**) for Hypernym Traversal.





Analysis

RESULTS ANALYSIS

Gaussian Distribution for Correct Predictions on Term Lists from direct synset

Dataset prepared only from synset directly	Single Support Mean and Standard Deviation(SD)	Multiple Support Mean and Standard Deviation(SD)
Word2Vec	Mean = 0.38 SD = 0.19	Mean = 0.39 SD = 0.18
fastText	Mean = 0.55 SD = 0.15	Mean = 0.55 SD = 0.13
Glove	Mean = 0.27 SD = 0.18	Mean = 0.26 SD = 0.16

Gaussian Distribution for Incorrect Predictions on Term Lists from direct synset

Dataset prepared only from synset directly	Single Support Mean and Standard Deviation(SD)	Multiple Support Mean and Standard Deviation(SD)
Word2Vec	Mean = 0.15 SD = 0.09	Mean = 0.18 SD = 0.09
fastText	Mean = 0.33 SD = 0.11	Mean = 0.39 SD = 0.10
Glove	Mean = 0.11 SD = 0.099	Mean = 0.11 SD = 0.094

Analysis

RESULTS ANALYSIS

Gaussian Distribution for Correct predictions on Term Lists from Hypernym Traversal

Dataset prepared from Hypernym Traversal on WordNet	Single Support Mean and Standard Deviation(SD)	Multiple Support Mean and Standard Deviation(SD)
Word2Vec	Mean = 0.30 SD = 0.16	Mean = 0.36 SD = 0.16
fastText	Mean = 0.46 SD = 0.12	Mean = 0.55 SD = 0.12
Glove	Mean = 0.21 SD = 0.15	Mean = 0.27 SD = 0.15

RESULTS ANALYSIS

Gaussian Distribution for Incorrect Predictions on Term Lists from Hypernym Traversal

Dataset prepared from Hypernym Traversal on WordNet	Single Support Mean and Standard Deviation(SD)	Multiple Support Mean and Standard Deviation(SD)
Word2Vec	Mean = 0.15 SD = 0.09	Mean = 0.18 SD = 0.09
fastText	Mean = 0.36 SD = 0.09	Mean = 0.41 SD = 0.09
Glove	Mean = 0.10 SD = 0.094	Mean = 0.12 SD = 0.10

Three things we observe:

Observation

1. Multiple support makes similarity scores higher. This can be observed from shift in the Gaussian curve towards right from single support to multiple support. (For finding broader relations, it is helpful to have more words).

2. FastText performed better, but most of space in fastText is sparse[4], because the gaussian distribution has higher mean for this model for both correct and incorrect predictions. Hence, fastText model learns word-vectors positioned at high density area of space. Notice, however the difference among two means is as significant as other models.

3. Hypernym relations have lower mean and this was expected as they share lesser similarity than facet terms of direct synset.

Can we get better?

Observation

We observe that state of the art word-embeddings model provides great performance in recognizing Facet Terms against random words with high accuracy. However, when we look closer, the similarity distribution has significantly lesser mean.

Are Cup and Coffee associated or similar words? Does association imply similarity[4]? We need to draw a stricter boundary in word-embeddings space among similar and associated words.

The best performing model is not good enough for the task of facet Term recommendation. A good model for this task, would have mean centered around close to 1 and a very small standard deviation. If we can obtain, this model the algorithmic support of word-embeddings could be extended for the tasks of facets generation.

Idea of *relative* Co-occurrence

I eat *apples* after working out.

1. The relative position of ***eat*** with respect to ***apples*** is one before it.
2. Anything that can be substituted for ***apples*** also belongs to the same Facet - *the eatable category.*

To draw that stricter boundary among associated and similar words, between **substitutional similarity** and **distributional similarity**.

If two words X and Y generate the same positional context during training, then X and Y should have similar encoding in the word-vector representation.

Idea of *relative* Co-occurrence

I eat apples after working out.

oranges

X

$W_{i-2}, W_{i-1}, W_{i+1}, W_{i+2}$

Y

Bases on this idea, of relative co-occurrence, we plan to train a Skip Gram word-embeddings model, based on premise that similar words would generate similar positional context and as a consequence of this, will have higher similarity scores than associated words.

And such a word-embedding model, we believe would be suitable for Facet Term Recommendation Tasks.

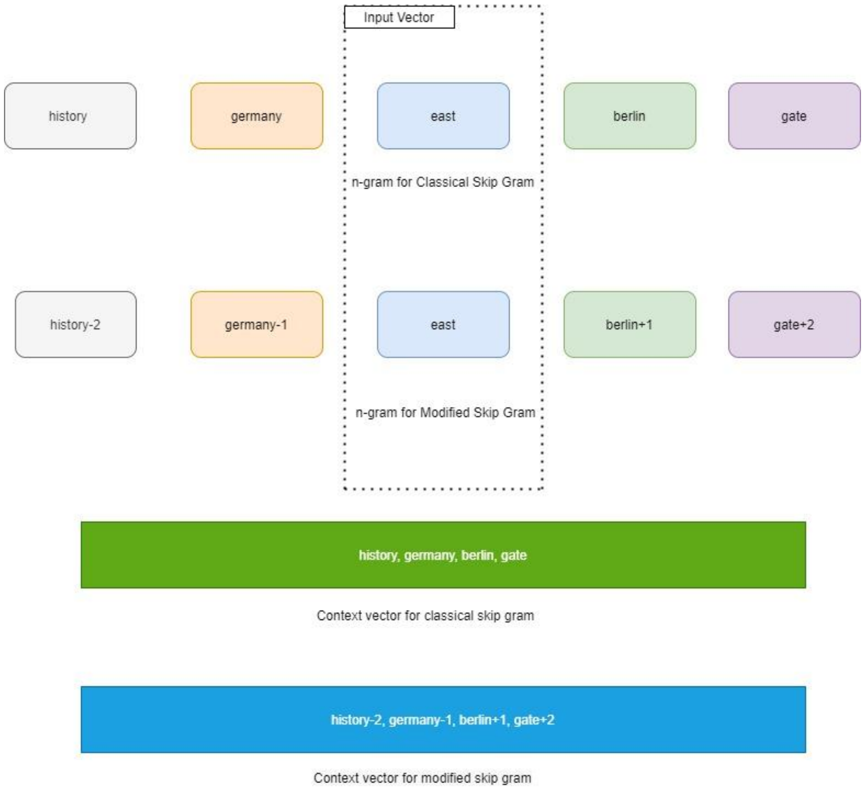
05 Developing Embeddings Model for Facet Generation Tasks

We shall develop two Skip Gram Models - Classic and **Modified**. The **Modified** Skip gram architecture shall accommodate positional context into training and is based on premise that similar words shall generate similar positional context.

We shall train the two models on similar criteria to eliminate any unfair advantage of one model over another. The criterias are:

1. Negative Sampling
2. Equal number of Epochs (50)

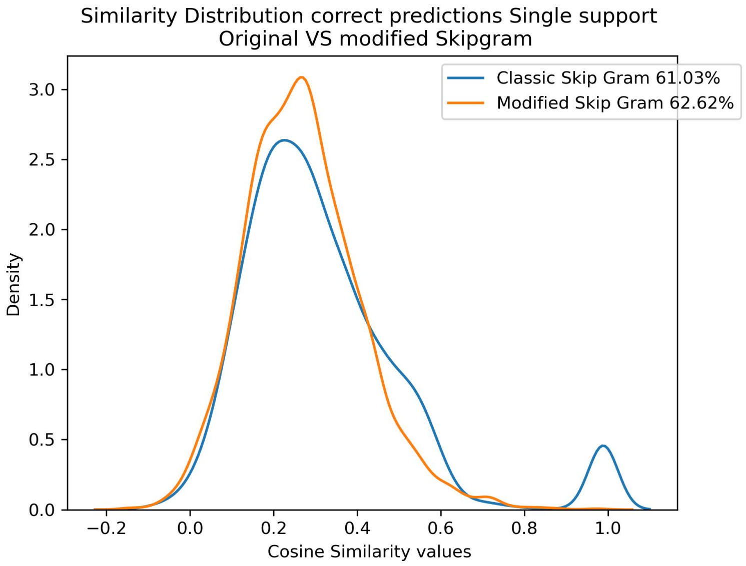
Modified Skip Gram Architecture

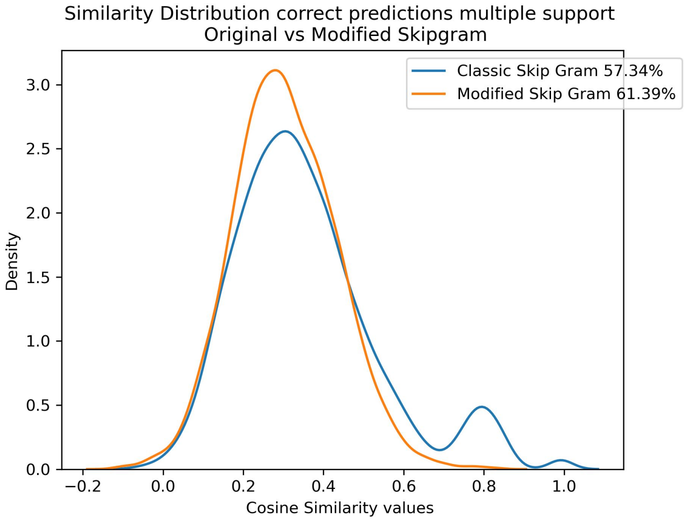


In the **Modified** Skip Gram Architecture, we take the positional co-occurrence in to the account.

Similarity Distribution of Correct Predictions on Term Lists from Direct Synset

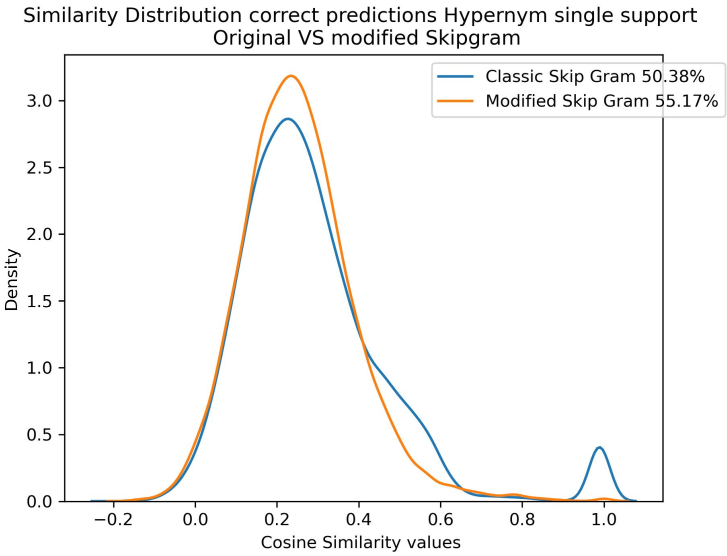
Analysis

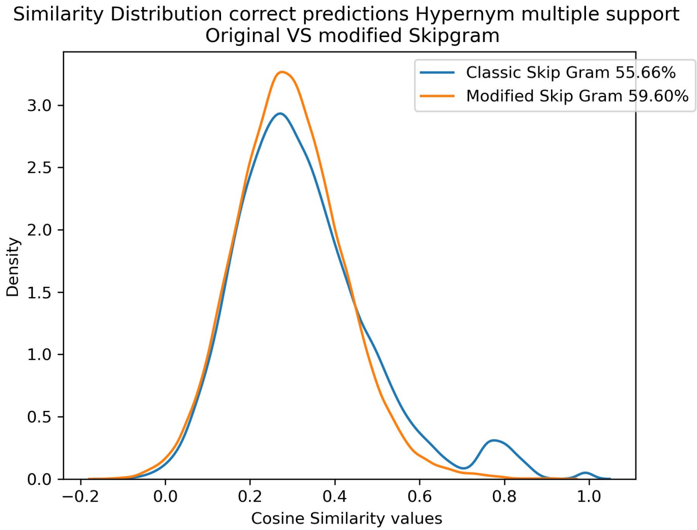




Similarity Distribution of Correct Predictions on Term Lists from Hypernym Relations

Analysis





Summary of Correct Predictions

Dataset prepared only from synset directly	Single Support Mean and Standard Deviation(SD)	Multiple Support Mean and Standard Deviation(SD)
Classic Skip Gram	Mean = 0.31 SD = 0.19	Mean = 0.35 SD = 0.17
Modified Skip Gram	Mean = 0.27 SD = 0.13	Mean = 0.30 SD = 0.12

Dataset prepared from Hypernym Traversal on WordNet	Single Support Mean and Standard Deviation(SD)	Multiple Support Mean and Standard Deviation(SD)
Classic Skip Gram	Mean = 0.28 SD = 0.18	Mean = 0.32 SD = 0.16
Modified Skip Gram	Mean = 0.25 SD = 0.13	Mean = 0.30 SD = 0.12

Summary of Incorrect Predictions

Dataset prepared only from synset directly	**Single Support Mean and Standard Deviation(SD)**	**Multiple Support Mean and Standard Deviation(SD)**
Classic Skip Gram	Mean = 0.30 SD = 0.21	Mean = 0.30 SD = 0.19
Modified Skip Gram	Mean = 0.23 SD = 0.11	Mean = 0.28 SD = 0.11

Dataset prepared from Hypernym Traversal on WordNet	**Single Support Mean and Standard Deviation(SD)**	**Multiple Support Mean and Standard Deviation(SD)**
Classic Skip Gram	Mean = 0.27 SD = 0.21	Mean = 0.30 SD = 0.18
Modified Skip Gram	Mean = 0.23 SD = 0.12	Mean = 0.27 SD = 0.11

06 Conclusion

1. The standard Deviation decreases across both correct and incorrect predictions, pointing out that **Modified** skip gram brings all the words to high density of space and is unable to draw a stricter boundary between similar and associated words.

2. The parameter space for word-embeddings for modified skip-gram is 4 times more than for the classic skip-gram and hence, it might need more training to give some efficient results.

The poor performance could be because of time-constraints(50 epochs), or we need way more data or perhaps something we are missing in our consideration.

Future Works

1. The conversions of semantic networks into semantic spaces has gained momentum recently and the word-embeddings models so obtained have performed substantially better than word-embeddings models trained on large collections of texts for semantic similarity tasks. Node Embeddings node2Vec to encode existing lexical graph structures through graph embedding Techniques[3].

2. We should dig further in drawing a stricter boundary between associated and similar words in Word-Embeddings models and other such efforts like substitutional similarity vs distributional similarity should be made in architectures for word-embeddings models.

Thank you for your attention!

[1] Feddoul, L., Schindler, S., & Löffler, F. (2019, September). Automatic Facet Generation and Selection over Knowledge Graphs. In International Conference on Semantic Systems (pp. 310-325). Springer, Cham.

[2] Tzitzikas, Y., Manolis, N., & Papadakos, P. (2017). Faceted exploration of RDF/S datasets: a survey. Journal of Intelligent Information Systems, 48(2), 329-364.

[3] Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings.

[4] Elekes, Á., Schäler, M., & Böhm, K. (2017, June). On the various semantics of similarity in word embedding models. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (pp. 1-10). IEEE. [Abstract], 1, 2.4, 4.5

[5] Medelyan, O., Witten, I. H., Divoli, A., & Broekstra, J. (2013). Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(4), 257-279. [Pre-requisites], 2, 3.5, 3.5.2