A distributional analysis
of affixal rivalry between *-ed* and *-y* in English

Ngo Trung Hieu
Zheng Peng *

Abstract

The paper *Affixal rivalry and its purely semantic resolution among English derived adjectives* by Nagano (2022) has claimed an interesting finding about the affixal rivalry between the ending -ed and -y in English. According to Nagano, there exists a purely semantic affixal rivalry between these two endings in the denominalized adjectives, created from English concrete nouns. The authors of this study is interested in looking at this finding from a different perspective, by using the distributional methodology, as demonstrated in Naranjo and Bonami (2023).

1 Literature review

1.1 Affixal rivalry in English affixes *-ed* and *-y*

1.2 Distributional methodology for affixal rivalry

1.3 Semantic retrofitting for word embeddings

2 Research questions

Because Nagano (2022) explained the affixal rivalry between *-ed* and *-y* in English from a purely semantic perspective, her experiments were qualitative, and therefore, the dataset used in the paper is very limited. We are interested in investigating this finding from a distributional perspective. Therefore, the research questions of this study are as follows:

- Is there similarity between the affixes *-ed* and *-y* in terms of their distributional behavior? To answer this question, this study will look at the distributional representations of the two affixes, by using conventional distributional word embeddings such as word2vec.

---

*M1 Linguistique Informatique - Université Paris Cité

- Are there evidence of affixal rivalry between *-ed* and *-y* in English, in a larger and not hand-picked dataset? To answer this question, this study will train a neural network on the dataset from Nagano (2022), and test the classifier on unseen data extracted from UniMorph. If there exists a rivalry between the two affixes, the classifier should not be able to accurately predict the correct affix for the unseen data.

- The conventional distributional representations of words may not be able to capture sematic relationships between words. Therefore, this study will also implement a retrofitting algorithm from Faruqui (2015) to augment the word embeddings. If the similarity between the two affixes is increased after retrofitting, it means that there exists a semantic relationship between the two affixes.

3   Methodology and Experiment design

There are three experiments in this project. To answer the first research question, we calculate and compare the difference vectors between the two affixes to see if they are semantically comparable or not. The second experiment is to train a neural network classifier on these difference vectors, then test the classifier on unseen data, in this case the data from UniMorph. The last question augmenting the semantics of the word embeddings using the semantic retrofitting method will be addressed in the third experiment.

3.1   Experiment 1:

For this experiment, the dataset from Nagano (2022) and a distributional word vectorization system are needed. The idea is to represent the roots of chosen words and their derivations in a vector space, so that we can use computation to calculate the similarity. The chosen vector space is the Google-News-300 dataset from word2vec model, which is accessible from the Gensim package. This vector space is created from a vocabulary of 3 million words from Google News, and each word is represented by a 300-dimensional vector. The dataset from Nagano (2022) is a hand-picked dataset of 38 words, which have both *-ed* and *-y* derivations. The roots of these words are concrete nouns, and the derivations are adjectives, so the process is denominalization. The difference vectors between the roots and the derivations are calculated by subtraction, and the average difference vectors of the two affixes are compared using cosine similarity. If the two affixes are distributionally similar, the difference vectors should be similar as well.

3.2   Experiment 2:

For the second experiment, we will create a neural network classifier with one hidden layer to train on the Nagano dataset, then use the classifier to predict the derivation of the unseen root words during training. The test dataset is extracted from UniMorph, and the words are filtered to only include words that can have *-ed* or *-y* affixes. The classifier will be trained

on the vectorized roots and their derivations from the Nagano dataset, and the classifier will be trained for 100 epochs using 10 folds to maximize learning efficiency due to the limited number of training examples. After training, the classifier will be tested on the UniMorph dataset of 4191 words, which were then vectorized by the word2vec model. The ratio of *-ed* affix is 42.92%, while the ratio of *-y* affix is 57.08%. The classifier will be tested on this UniMorph dataset, and the accuracy will be calculated. If the classifier can predict the derivation of the unseen words with an accuracy of around 50%, it means that the classifier is not able to distinguish between the two affixes, and therefore there exists an affixal rivalry between the two affixes.

3.3 Experiment 3:

Semantic retrofitting is a method to improve the quality of the word vectors by using the semantic information (synonyms, antonyms, hyper/hyponyms) from a lexicon.

In this experiment, we will use the retrofitting method from Faruqui et al. (2015) and the WordNet lexicon to improve the quality of the word vectors from the training data. The WordNet lexicon provides the semantic information for the words, and then use the retrofitting method to create a new word vector based on the old word vector and the "neighbors" of the word. The idea of retrofitting is to have a new word vector that is close (has a better cosine similarity) to the old word vector, but is also closer to the neighbors of the word. This new word vector is then said to be capturing the semantic information of the word better than the purely distributional word vector.

Then we will recalculate the difference vectors to see if there are improvements or not, and if these improvements are significant. If the difference vectors are more similar after retrofitting, it means that the two affixes are semantically related.

4 Results

4.1 Experiment 1:

Using the word2vec model, we first calculate the average similarity between two random words in the lexicon. To do this, we randomly chose 100,000 pairs from the lexicon, and compute the cosine similarity between the pairs. The result of this computation is 0.1308, and this serves as a baseline for the cosine similarity results.

The next step is to calculate the average difference vectors between the roots and the derivations of the Nagano dataset. The vectorization model is used on both the roots and the derivations of the words in the Nagano dataset, and then the difference vector is calculated as difference vector = root vector - derivation vector. The difference vectors between each root and its two derivations are then compared using cosine similarity, then these values were used to calculate the average similarity between the two affixes *-ed* and *-y*. The result of this computation is 0.4162, which is 3.18 times higher than the baseline. This result shows that

the two affixes are distributionally similar. The detailed result of this experiment is shown in Table 1.

4.2   Experiment 2:

First, the test data was extracted from the UniMorph database, only finding the denominalized adjectives ending in -*ed* and -*y*. This results in a list of 4191 words, with 42.92% ending in -*ed* and 57.08% ending in -*y*.

Next, a neural network classifier was initialized using PyTorch. This classifier has one hidden layer of 128 neurons, using ReLU as the activation function. The input layer has 300 neurons, which is the same as the dimension of the word vectors. The output layer has 2 neurons, which is the number of classes.

The classifier was then trained on the Nagano dataset of 76 examples. The classifier was trained for 100 epochs, using 10 folds to maximize learning efficiency. When tested on the test dataset from UniMorph, the classifer gave an accuracy of 49.146%, which is slightly lower than the expected result of 50%. This result shows that the classifier trained on the difference vectors of -*ed* and -*y* is not able to distinguish between the two affixes, and therefore distributionally, there exists an affixal rivalry between these two affixes. The detailed result of this experiment is shown in the attached Jupyter notebook. However, this result is still lower than the most frequent class of 57.08%, so there might be room to improve.

4.3   Experiment 3:

For the last experiment, we wanted to semantically retrofit the word vectors to see if there is semantic similarity between -*ed* and -*y* or not. A lexicon created from WordNet was used to provide the semantic information for the words. The retrofitting method from Faruqui et al. (2015) was used to create a new word vector based on the old word vector and the semantic information from the lexicon. The new word vector is calculated as follows: new word vector = α * old word vector + $\sum_{i=1}^{d_i} \frac{1}{d_i}$ * (neighbor vectors), where α is a hyperparameter and d_i is the number of neighbors of the word. The value of α is set to 1. This algorithm allows the new word vector to not only be similar to the old word vector, but also more similar to the semantically related neighbors of the word.

The retrofitting method was applied to the word vectors of the Nagano dataset, and then the difference vectors were recalculated. The average similarity between the two affixes was then calculated again, and the result is 0.495. Compared to the original vectors, the average similarity was increased only by a small amount of 0.0788, but it is still 3.19 times higher than the baseline. This modest gain was due to the fact that some of the similarities of the word vectors and their derivations were increased substantially,, such as *bone* from 0.602 to 0.998 or *curve* from 0.628 to 0.997, a few others were decreased. However, overall, the average similarity was increased, which shows that the two affixes are semantically similar. The detailed result of this experiment is shown in Table 2.

ROOT	COSINE_SIMILARITY
leg	0.526988
bone	0.602122
price	0.537156
wit	0.264876
curve	0.628152
head	0.354714
brain	0.603593
cheek	0.448141
feather	0.641470
hand	0.227643
head	0.354714
hip	0.482956
mouth	0.506514
nose	0.364511
skin	0.517622
tooth	0.620628
edge	0.395138
loft	0.324167
room	0.246554
shape	0.319353
taste	0.323992
fish	0.202100
leaf	0.354984
rock	0.445634
sand	0.314394
stone	0.345706
water	0.352573
air	0.181804
cloud	0.606249
dust	0.459283
ice	0.322265
mist	0.297710
snow	0.462827
sun	0.424234
dress	0.036261
oil	0.509179
spice	0.556888
sugar	0.652546

Table 1: Cosine similarity between the roots and their derivations

ROOT	DERIV_ed	DERIV_y	COSINE_SIMILARITY	OLD_COSINE_SIMILARITY
bone	boned	boney	0.998281	0.602122
curve	curved	curvy	0.997423	0.628152
head	headed	heady	0.996630	0.354714
brain	brained	brainy	0.996457	0.603593
hip	hipped	hippy	0.482956	0.354714
mouth	mouthed	mouthy	0.506514	0.482956
nose	nosed	nosy	0.364511	0.506514
skin	skinned	skinny	0.517622	0.364511
tooth	toothed	toothy	0.998461	0.517622
edge	edged	edgy	0.395138	0.620628
loft	lofted	lofty	0.324167	0.395138
room	roomed	roomy	0.246554	0.324167
shape	shaped	shapely	0.996835	0.246554
taste	tasted	tasty	0.323992	0.319353
fish	fished	fishy	0.202100	0.323992
leaf	leafed	leafy	0.354984	0.202100
rock	rocked	rocky	0.445634	0.354984
sand	sanded	sandy	0.314394	0.445634
stone	stoned	stony	0.345706	0.314394
water	watered	watery	0.352573	0.345706
air	aired	airy	0.181804	0.352573
cloud	clouded	cloudy	0.606249	0.181804
dust	dusted	dusty	0.459283	0.606249
ice	iced	icy	0.322265	0.459283
mist	misted	misty	0.297710	0.322265
snow	snowed	snowy	0.462827	0.297710
sun	sunned	sunny	0.424234	0.462827
dress	dressed	dressy	0.036261	0.424234
oil	oiled	oily	0.509179	0.036261
spice	spiced	spicy	0.556888	0.509179
sugar	sugared	sugary	0.652546	0.556888

Table 2: Original and retrofitted cosine similarities.

5   Discussion

The results of the three experiments show that there exists both distributional and semantic similarity between the two affixes *-ed* and *-y*. However, the accuracy of the second experiment is still much lower than the baseline of predicting the most frequent affix, which is 57.08%. We looked at this problem from a few other perspectives:

- The classifier might be learning too much unnecessary details from the train set due to the power of neural network, therefore cannot generalize well to the test set. To solve this problem, we implemented an early stopping mechanism to stop the training process when the test accuracy stops improving. With the early stopping mechanism implemented, the classifier had the accuracy for the train set at 100% in just one epoch, and the training was stopped because the accuracy for the test set was decreased in the next epoch. The accuracy for the test set was 56.662%, which is much higher than the previous result but still a little lower than the most frequent baseline. This result showed that the classifier is indeed learning too much unnecessary details from the train set, and the early stopping mechanism is able to solve this problem.

- The training set might be too small for the classifier to learn the difference between the two affixes. To solve this problem, we reversed the train set and test set, so that the train set is substantially larger than the test set. After the reversal, the accuracy of the prediction on the Nagano set was at exactly 50%, which is the baseline for this dataset. This result showed that the datasets were not the problem.

- The train set and the test set might be too different from each other for the classifier to work, or the classifier is not efficient enough to capture information. To check this issue, we implemented the classifier on another task of predicting if a word is in singular or plural form. The data was extracted from UniMorph, and splitted into a train and test set with the ratio of 80:20. After training, the classifier achieved an accuracy score of 45.45% on the test set, which is a little lower than the baseline of 50%. This result showed that the classifier is not efficient enough to capture information, and the train set and test set are not too different from each other.

After looking closer into the problem from other perspectives, all of the above are not the case. From this, we can conclude that there is indeed affixal rivalry between *-ed* and *-y*, and the two affixes are also semantically similar.

6   Conclusion

This study attempted to distributionally prove the findings from Nagano's paper in 2022, that there exists a semantically affixal rivalry between the affixes *-ed* and *-y*. The distributional difference vectors of the two affixal process are cosine similar, and retrofitting these vectors with semantic information managed to improve the similarity, therefore the two process are semantically similar. The neural network classifier managed to achieve an accuracy

of 56.662% on the UniMorph dataset, which translates to the fact that a classifier cannot distinguish between the two affixes, and therefore there exists an affixal rivalry between the two affixes. From these experiments, we can conclude the findinds of Nagano in 2022, that indeed there exists a semantically affixal rivalry between the affixes *-ed* and *-y*.

References

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons, 2015.

Matias Guzmán Naranjo and Olivier Bonami. A distributional assessment of rivalry in word formation. *Word Structure*, 2023.

A Nagano. Affixal rivalry and its purely semantic resolution among english derived adjectives. *Journal of Linguistics*, pages 1–32, 2022. doi: 10.1017/S0022226722000147.