From Strings to Other Things: Linking the Neighborhood and Transposition effects in Word Reading

Stéphan Tulkens and **Dominiek Sandra** and **Walter Daelemans**
CLiPS - Computational Linguistics Group
Department of Linguistics
University of Antwerp
{stephan.tulkens,dominiek.sandra,walter.daelemans}@uantwerpen.be

Abstract

We investigate the relation between the transposition and deletion effects in word reading, i.e., the finding that readers can successfully read "SLAT" as "SALT", or "WRK" as "WORK", and the neighborhood effect. In particular, we investigate whether lexical orthographic neighborhoods take into account transposition and deletion in determining neighbors. If this is the case, it is more likely that the neighborhood effect takes place early during processing, and does not solely rely on similarity of internal representations. We introduce a new neighborhood measure, rd20, which can be used to quantify neighborhood effects over arbitrary feature spaces. We calculate the rd20 over large sets of words in three languages using various feature sets and show that feature sets that do not allow for transposition or deletion explain more variance in Reaction Time (RT) measurements. We also show that the rd20 can be calculated using the hidden state representations of an Multi-Layer Perceptron, and show that these explain less variance than the raw features. We conclude that the neighborhood effect is unlikely to have a perceptual basis, but is more likely to be the result of items co-activating after recognition. All code is available at: www.github.com/clips/conll2018

1 Introduction

Despite their many disagreements and differences, a common thread among many models of word reading is that they attempt to explain differences in reading speeds by assuming that similarity between words modulate reading speed. There is good reason for this assumption; many experiments have shown that responses on trials are modulated by a word's similarity to other words, be it semantic (Rodd et al., 2002, 2004), orthographic (Andrews, 1997; Perea and Pollatsek, 1998), or phonological similarity (Van Orden, 1987; Rastle and Brysbaert, 2006).

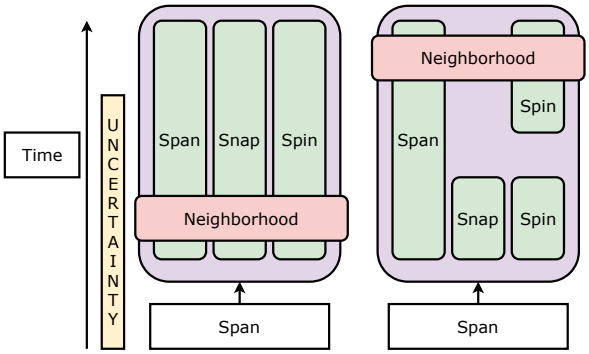

Figure 1: This diagram shows the two positions contrasted in this paper. The left model is the early model, in which the neighborhood effect arises before perceptual uncertainty is resolved; this causes transposition and substitution neighbors to count as neighbors. In the late model, the neighborhood effect only arises after perceptual uncertainty is resolved, and transposition and substitution neighbors do not count towards the neighborhood.

In psycholinguistic research on word reading, this has led to the common practice of including a measure of orthographic neighborhood similarity as a control variable, as these neighborhood measures explain variance in word reading even when controlling for frequency and length (Coltheart, 1977; Yarkoni et al., 2008).

Orthographic neighborhood measures are usually operationalized using edit distance metrics, such as the Levenshtein distance (Levenshtein, 1966). The most well-known measure of neighborhood size is Coltheart's N (Coltheart, 1977), which is the number of types within a substitution distance of 1. Yarkoni et al. (2008) show that N is nearly always 0 for longer words, as long words tend are less frequent, and present an alternative to N, called old20, which is the mean Levenshtein distance to the 20 closest neighbors. old20 corre-

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 75–85
Brussels, Belgium, October 31 - November 1, 2018.

lates well with reaction time (RT) measures on two experiments, and explains more variance than N after accounting for length and frequency, and is therefore considered superior to N, and often used as a *de facto* replacement for N (Yarkoni et al., 2008).

Despite its ubiquity as a control variable, the cause for the neighborhood effect is unknown or disputed (Perea, 2015). One aspect, which we explore in the current work, is that it is currently unknown whether the neighborhood effect is *early* or *late*. If the neighborhood effect is *early*, it is caused by the visual stimulus co-activating multiple representations. If it is *late*, the effect is caused by an already activated representation co-activating similar representations.

Of particular interest regarding this question is the finding that skilled readers are remarkably proficient in reading words that contain transposed letters, e.g. "SLAT" versus "SALT" (Davis and Bowers, 2006; Grainger, 2008), or words from which letters are deleted, e.g. "WRK" and "WORK" (Schoonbaert and Grainger, 2004). In this work, we refer to these two effects in tandem as *flexible letter encoding*. Examples of models that try to explain flexible letter encoding include the open bigram family of models (Whitney, 2001; Grainger and Van Heuven, 2004; Schoonbaert and Grainger, 2004; Whitney and Cornelissen, 2008), the SOLAR model (Davis, 2001), the overlap model (Gomez et al., 2008), and, most recently, the spatial coding model (Davis, 2010b).

Taking into consideration both the neighborhood effect and flexible letter encoding, we define the following research question: are the neighborhoods also defined using flexible letter encoding? That is, if we know that readers activate "SALT" upon reading "SLAT", does this also imply that the lexical neighborhood of "THREE" includes "THERE"?

To answer this question, we calculate the neighborhood density using a variety of feature sets, including features that do not allow for flexible letter encoding, and those that do. If lexical neighborhoods calculated using flexible letter encodings account for less variance in word reading times than neighborhoods based on slot-based features, we can surmise that it is more likely that the neighborhood effect is late in origin. This follows from the fact that flexible letter encodings are most likely to be an intermediate encoding step towards a concrete internal representation. Hence, if neighborhoods with flexible letter encodings explain less variance, flexible letter encoding most likely does not play a role in the neighborhood effect. This, in turn implies that the neighborhood effect is likely a late effect, and is caused by concrete representations co-activating similar representations. These two positions are contrasted in Figure 1.

2 Main Contributions

To quantify the effect of different forms of representations and their respective neighborhoods, we introduce the Representation Distance 20 (rd20), a generalization of old20 which operates on arbitrary feature spaces.

We first replicate the original findings of Yarkoni et al. (2008) regarding old20 and N on Dutch, British English, and French lexical databases. As old20 uses the Levenshtein metric, which encodes flexible letter position by allowing deletions and transpositions, the neighborhoods defined by old20 in principle support the idea of flexible letter encoding.

Comparing to old20 and N, we show that rd20 can be used to create neighborhood measures for various feature sets. Furthermore, we use regression models to quantify the relation between word length, frequency and rd20 on the one hand, and Reaction Times (RT) in lexical decision experiments on the other. We do this for four different feature sets on all aforementioned lexical databases. Two of the four feature sets are slot-based feature sets used in older models of computational psycholinguistics, and two of them are used by models that assume some kind of flexible letter encoding. We can therefore use rd20 to assess the effect of the representational assumptions in models of flexible letter encoding, as well as provide a direct comparison to old20.

We show that rd20 using one hot encoded letter features explains slightly more variance in lexical decision experiments than old20. The fact that rd20 takes much less time to compute and is more flexible in the choice of inputs shows that it is a practical alternative to old20. Additionally, we show that the rd20 of feature sets which specifically encode letters in a flexible manner explains far less variance in RT than the rd20 of encodings which do not support flexible letter encoding. This leads us to hypothesize that lexical neighbor-

hoods are not defined using flexible letter encoding, and that, consequently, the neighborhood effect itself is a late effect, that is, an effect caused by co-activation of similar representations, and not caused by the visual likeness of stimuli.

To provide additional evidence regarding the statement that the neighborhood effect follows from internal representations, we perform an experiment using Multi-Layer Perceptrons. After training the network on each feature set, we calculate rd20 of the hidden states of these networks, and use these distances as a predictor in a linear regression experiment

2.1 Representation Distance 20

Representation Distance 20 (rd20) is a measure that does not assume a particular representational format, and thus applies to any kind of vector representation. It is therefore well-suited to quantifying the effect feature sets have on lexical decision experiments.

The rd20 for a featurized word x given a set of featurized words X, where $x \in X$, is defined as follows:

$$s(x, X) = sort(cos(x, X)) \tag{1}$$

Where $sort$ is a sorting operator, cos is the cosine distance, x is featurized item, and X is the set of featurized items. We then take the mean of the 20 first items, excluding the item itself.

$$\text{rd20}(x, X) = \frac{\sum_{i=1}^{21} s(x, X)^i}{20} \tag{2}$$

We use the 20 closest neighbors to be able to compare to old20, which also uses 20 neighbors. As Yarkoni et al. (2008) note, the value of 20 is quite arbitrary, and values between 5 and 50 seem to work well for most experiments. Because rd20 uses the cosine distance, it directly applies to any vector representation. It is therefore suitable for inspecting both external phenomena, i.e. featurized string representations, and internal representations, e.g. weight matrices of neural networks.

3 Materials

This section describes the materials used in the paper: the corpora, reaction time datasets, and the various feature sets.

3.1 Corpora

Throughout the paper we use three different lexical databases derived from subtitle corpora as the source of our words and frequency counts. For Dutch we use SUBTLEX-NL (Keuleers et al., 2010a), for English we use SUBTLEX-UK (Van Heuven et al., 2014), and for French we use Lexique 3 (New et al., 2007). Frequency counts from subtitle corpora account for substantially more variance in Reaction Time measurements, and are based on far larger corpora, than previously available databases (Brysbaert and New, 2009; Brysbaert and Cortese, 2011), such as CELEX (Baayen et al., 1993) and previous versions of Lexique (New et al., 2001).

For all three languages, we use reaction times (RT) from megastudies (Seidenberg and Waters, 1989). For Dutch we extract the reaction times from the Dutch Lexicon Project 2 (DLP) (Keuleers et al., 2010b; Brysbaert et al., 2016), for English we use the British Lexicon project (BLP) (Keuleers et al., 2012), and for French we use the French Lexicon project (FLP) (Ferrand et al., 2010). As with the subtitle corpora, these megastudies provide us with a more accurate estimate of Reaction Times than previous studies with a smaller number of participants and a smaller set of items.

We extract a subset of these corpora according to the following procedure: for each language, we take all words from the SUBTLEX corpora and lexicon projects, removing any words which were shorter than 2 characters, or words which contained non-alphabetic characters, such as '#' and '-'. We then remove any words from the lexicon project database which are not in the SUBTLEX database, such that the words extracted from the lexicon project were a subset of those in the SUBTLEX database.

Additionally, for all languages we remove any diacritic markers, transforming e.g. the French word 'très' to 'tres'. This was done because not all feature sets can appropriately featurize these diacritic markers.

For each language, this leaves us with a set of SUBTLEX words, for which we only have frequency counts, and a set of words from the lexicon project, for which we have both frequency counts and Reaction Time measurements. The sizes of the resulting corpora are listed in Table 1.

	Dutch	English	French
SUBTLEX	117,789	157,378	115,550
Lexicon project	24,908	28,530	36,677

Table 1: The number of words left over in the SUBTLEX and Lexicon projects after filtering. Note that we removed any words from the Lexicon project which were not in the SUBTLEX database, so that the words from the lexicon project are an exact subset of those in the SUBTLEX database.

3.2 Features

We use four different orthographic feature sets. All the feature sets were previously implemented in `wordkit` (Tulkens et al., 2018).

3.2.1 Slots

The two slot-based feature encodings are created by left-justifying strings, padding them with spaces to the length of the longest word in our corpus, and then replacing each letter in each resulting slot by a feature vector. These feature vectors are then concatenated to create a final feature vector. As noted in the introduction, these types of encodings are thought to be unrealistic (Grainger and Van Heuven, 2004; Davis and Bowers, 2006), as they predict that words which are not aligned have low similarity. The words "STAR" and "TAR", for example, have a similarity of 0 according to a naive slot-based encoding. Despite this shortcoming, the influence of slot-based encodings on contemporary models of word reading can not be understated (Miikkulainen, 1997; McClelland and Rumelhart, 1981; Harm and Seidenberg, 2004; Coltheart et al., 2001).

One hot encoded characters One hot encoded character featurization assigns a single orthogonal vector to each character, and hence assumes that there is no underlying similarity, visual or otherwise, between letters. This encoding is closest to the encoding implicitly used by the Levenshtein distance, and used by old20. In this encoding we treat the space character as a separate character, and not as a zero vector.

Fourteen segment encoding The fourteen segment encoding was first introduced by Rumelhart and Siple (1974), and is used in the original version of the Interactive Activation model (McClelland and Rumelhart, 1981). As its name implies, it uses fourteen binary segments, each of which denotes a specific vertical, horizontal, or diagonal line segment. Because the encoding is subsymbolic, words with different letters in the same slot might still have some overlap in their similarity. In this encoding, we treat the space character as a zero vector.

3.2.2 Wickelgraphs

Wickelgraphs were first introduced as Wickelphones in the context of phonological representations (Seidenberg and McClelland, 1989) and are named after, and based on the work of, Wickelgren (1969). As we saw above, slot-based encodings predict that words which are not aligned are completely dissimilar. Wickelgraphs attempt to overcome this downside by representing words as sets of contiguous $n$grams, where n is usually set to 3, and $n-1$ padding characters are added to the start and end of each word. For example, the word "SALT" has the following wickelgraph representation: {##S, #SA, SAL, ALT, LT#, T##}.

3.2.3 Weighted Open bigrams

Another way of representing flexible letter coding in reading is the open bigram family of feature encodings. Open bigrams were first proposed by Whitney (2001) to account for readers' resilience to letter transposition effects, although earlier accounts of transposition-like encodings can be found in work by Mozer (1987). For a criticism of open bigrams, see work by Davis (2010a) and Kinoshita and Norris (2013).

Open bigrams are constructed by taking the ordered set of 2-combinations of all letters in a word. For example, the word 'SALT' becomes {SA, SL, ST, AL, AT, LT} in an open bigram encoding scheme. This scheme can account for transposition and deletion effects because most bigrams survive the transposition or deletion of two letters.

The weighted open bigram scheme attaches a weight to each bigram combination depending on the distance between the constituent letters of the bigram in the word (Schoonbaert and Grainger, 2004; Whitney and Cornelissen, 2008; Whitney, 2001). This encoding scheme was introduced to account for the observation that participants experience more inhibition to transpositions which are further apart. Following Whitney et al. (2012) we used weights of 1.0, .7, and .2 for bigrams with 0, 1, or 2 intervening letters in all our experiments. Bigrams with more than 2 intervening letters get a weight of 0, and are therefore discarded in the distance computation.

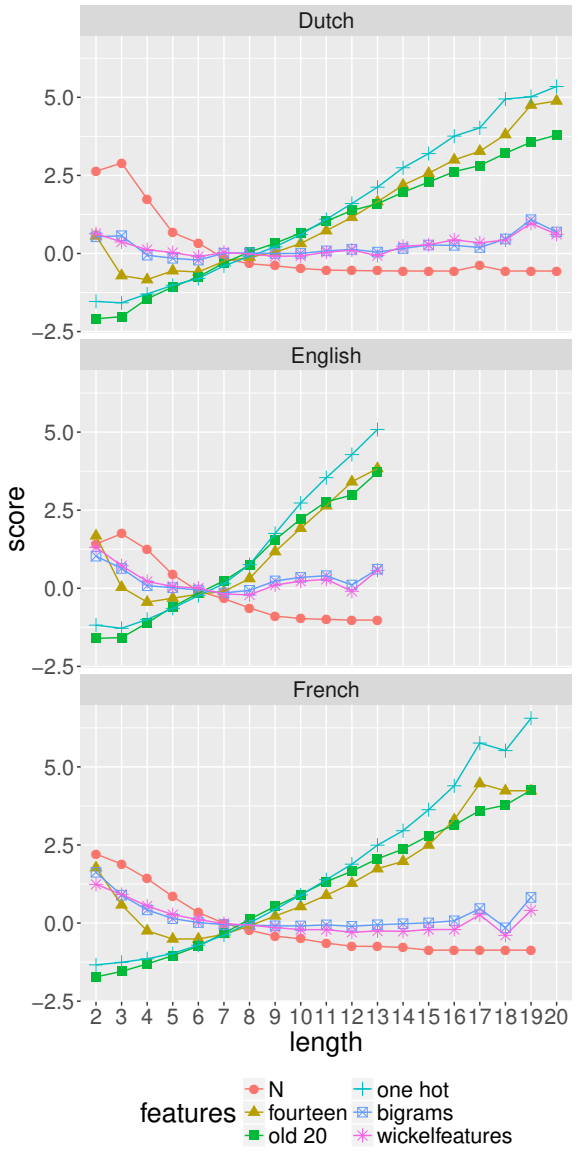

Figure 2: rd20, N and old20, plotted against word length for three languages. The figure shows the measures behave the same across languages. The y-axes denote the scaled quantities, as the old20 and N measures are expressed on a different scale than the various rd20 measures.

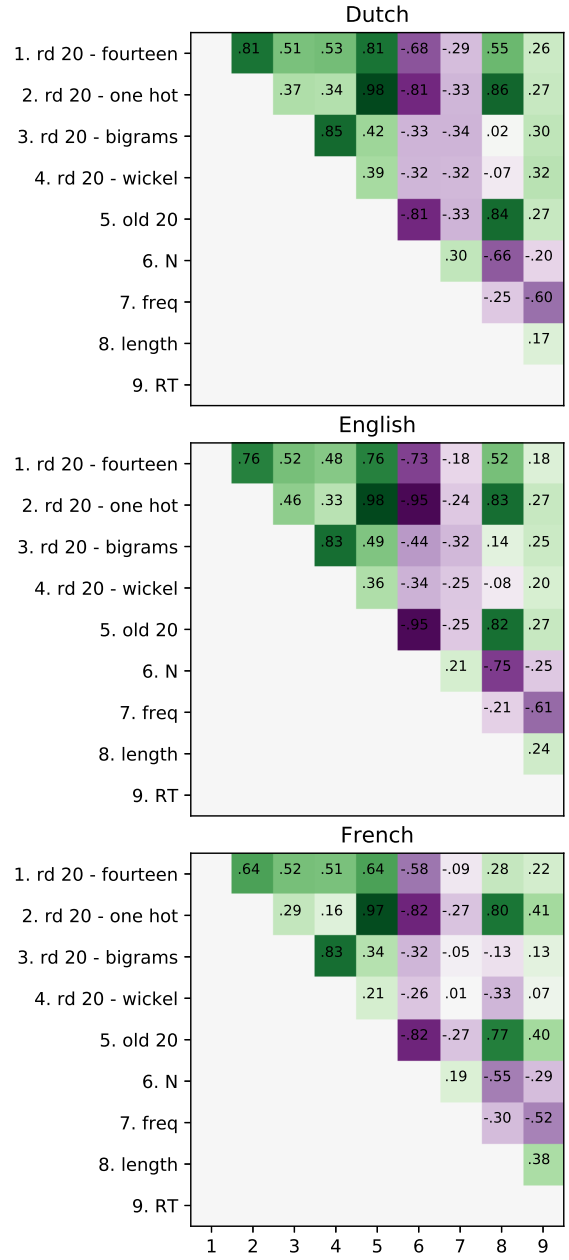

Figure 3: The correlations between the control variables (length and frequency), the various distance measures, and RT. All correlations are significant ($p < .05$).

4 Experiment 1: empirical validation of rd20

Using the materials defined in Section 3, we carry out comparative experiments of old20, N, and the rd20 of the four feature sets described above.

Figure 2 shows the word length versus the mean distance for each of the measures for all three languages. The figure shows that old20 and the measures based on slot-based encodings correlate strongly with length, while flexible encodings do not correlate with length. We observe the same pattern of performance for all three languages. As a similar pattern of performance was observed in Yarkoni et al. (2008), we consider this to be an empirical validation of our datasets.

Figure 3 shows the Spearman correlations between the different predictor variables (length, frequency), and the various measures for all languages. As the figure indicates, the pattern of correlations is consistent across all surveyed languages, and only differs in magnitude, not direction. Additionally, because the results corre-

		Dutch			English			French		
	Predictor	β	R^2_{adj}	ΔR^2_{adj}	β	R^2_{adj}	ΔR^2_{adj}	β	R^2_{adj}	ΔR^2_{adj}
base	length	.025	.252	.0	.083	.344	.0	.263	.314	.0
	freq	-.494			-.558			-.414		
rd20 - fourteen	length	-.058	.270	.018	.053	.346	.002	.209	.339	.025
	freq	-.472			-.553			-.421		
	score	.161			.059			.164		
rd20 - one hot	length	-.305	.292	**.040**	-.066	.353	**.011**	.003	.353	**.039**
	freq	-.459			-.552			-.417		
	score	.397			.181			.326		
rd20 - bigrams	length	.044	.273	.021	.079	.349	.005	.292	.342	.028
	freq	-.438			-.536			-.401		
	score	.154			.076			.167		
rd20 - wickel	length	.006	.289	.037	.096	.351	.007	.333	.350	.036
	freq	-.417			-.533			-.397		
	score	.206			.088			.200		
old20	length	-.240	.283	.022	-.051	.352	.008	.035	.349	.035
	freq	-.457			-.550			-.412		
	score	.329			.166			.295		
N	length	.087	.259	.007	.078	.344	.000	.261	.314	.0
	freq	-.510			-.557			-.414		
	score	.110			-.007			-.005		

Table 2: The coefficients, explained variance, and change in explained variance of the regression analyses. The rd20 measure using one hot features explains the most variance across all languages, although the difference is not significant for English.

spond with those from Yarkoni et al. (2008), this provides additional evidence for old20 and our datasets. Given that old20 is considered to be a good neighborhood measure, and the various rd20 measures show the same type of effects, i.e., effects in the same direction, this indirectly validates rd20 as a good measure.

As an aside, while we see the same direction of effects as in Yarkoni et al. (2008), we do see that the magnitude of the correlations between the scores and RT are lower for all corpora, which was reported to be .612 for the English Lexicon Project stimuli used in Yarkoni et al. (2008).

4.1 Regression analyses

In addition to the zero-order correlations above, we also conduct stepwise regression analyses. We use the RT values from the various lexicon projects, as explained in Section 3 as dependent variables, and consider the length, frequency, and the distance measures as independent variables. We first start by adding the control variables, length and frequency in this case. Then, for each defined measure, we add the score predictor as an additional variable, while measuring the effect this addition has on model fit.

The difference between the adjusted R-squared, or R^2_{adj} from here on, of the model with the control variables and the model with the extra predictor is called the ΔR^2_{adj}, and explains how much additional variance is explained by the added predictor. Because all measures were calculated using the same data, we can simply compare the ΔR^2_{adj} of each of the regression models to determine the effect of that particular measure.

The results of the regression analyses are shown in Table 2. The rows above the horizontal line show the base model, i.e. the model with only the control variables as predictors, while the rows below the line denote the various statistics of the different models with respect to the base model.

All score predictors for each model but the N model show positive effect of score on RT, indicating that words in denser neighborhoods, i.e. words with a *lower* average distance to nearest neighbors, have shorter Reaction Times. These scores thus predict a positive effect of neighborhood density.

For N we expect a negative correlation, as the

measure is inverted, i.e. words with denser neighborhoods have higher figures. Nevertheless we see a positive effect of N for Dutch, which is unexpected.

In all three corpora the one hot encoded features explain the most variance out of all the measures, with the wickelfeatures following in second place for Dutch and French, and OLD20 following in second place for English. To see if these difference were significant, we bootstrapped the difference between the R^2_{adj} estimates of one hot encoded rd20 and other feature sets with an α of .05. For Dutch, we bootstrapped the differences between the one hot encoded and wickelfeatures; which led to intervals of [0.0004, 0.0058], indicating a significant, albeit really small, difference between the one hot encoded and wickelfeatures. For English and French, we compared old20 to both the rd20 of the one hot and the wickelfeatures. Because of multiple comparisons, we used Bonferroni correction to correct our α of .05 to .025. For English, the confidence interval of the bootstrapped differences between the one hot encoding and wickelgraphs was [-0.0028, -.0003], indicating significance, while the same confidence interval for one hot encoding and old20 was [0.0003, 0.0020], again indicating a significant difference. For French, the confidence intervals for the differences between one hot encoding and wickelgraphs were [-0.0011, 0.0032], indicating a non-significant difference, while the confidence interval for the differences between one hot encoding and old20 was [0.0029, 0.0061], again indicating significance.

In a practical sense, the significance is not that important: as all of these values are really small, there seems to be little reason to prefer one of the metrics over the other. That is, even though the difference between old 20 and the rd20 of a one hot encoded representation is significant, the difference in explained variance is so small to not really matter.

Theoretically, these results point towards a smaller role for transposition effects than previously assumed, for two reasons:

First, given that the main difference between the one-hot encoded features and the Levenshtein-based old20 is that the Levenshtein metric allows for transpositions and deletions, we can view the difference in explained variance between these two measures as the *net transposition effect*. If

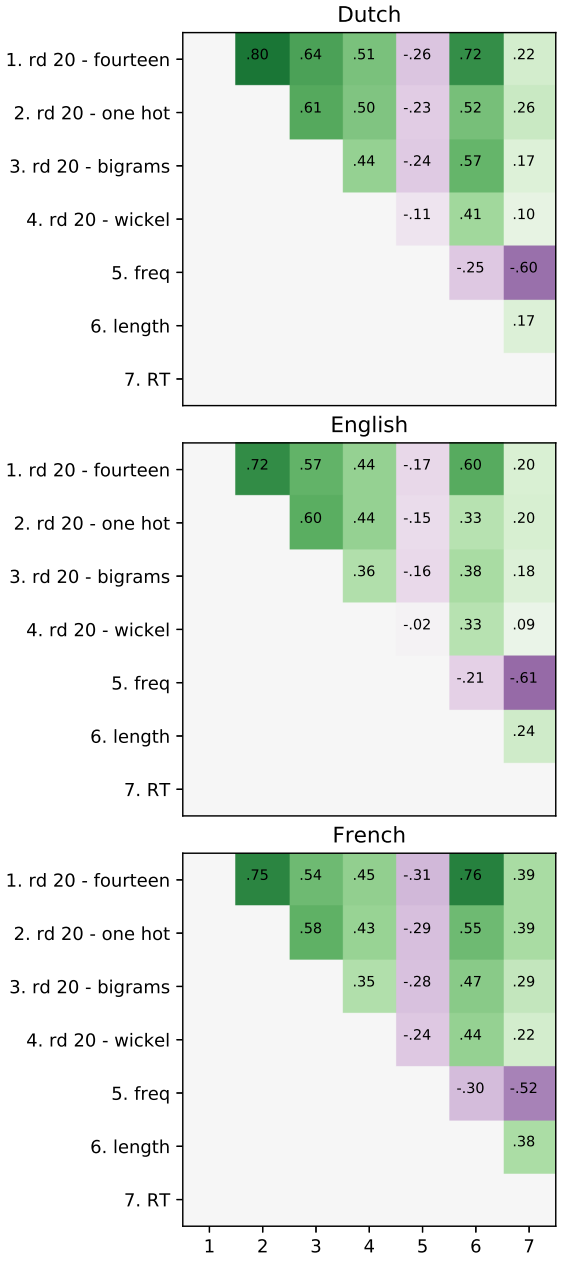

Figure 4: The correlations between the control variables, length and frequency, and the various distance measures for representations learned by the MLP.

transpositions and deletions played a large role during lexical access, then we would expect to see a large positive net transposition effect. In our experiments, we see exactly the opposite: a small but significant negative net transposition effect in all corpora. Second, we observe that the bigrams, the feature set specifically constructed for modeling transposition effects during word reading, explains less variance than the slot-based encodings in all cases.

Both of these results lead us to hypothesize that

		Dutch			English			French		
	Predictor	β	R^2_{adj}	ΔR^2_{adj}	β	R^2_{adj}	ΔR^2_{adj}	β	R^2_{adj}	ΔR^2_{adj}
base	length	.025	.252	.0	.08	.344	.0	.263	.315	.0
	freq	-.493			-.55			-.414		
fourteen	length	-.054	.257	.005	.031	.348	.004	.143	.327	.012
	freq	-.484			-.556			-.400		
	score	.110			.083			.168		
one hot	length	-.068	.274	**.022**	.041	.356	**.012**	.144	.347	**.032**
	freq	-.475			-.551			-.390		
	score	.179			.119			.221		
bigrams	length	.020	.252	.0	.061	.346	.002	.23	.319	.004
	freq	-.492			-.554			-.403		
	score	.008			.051			.075		
wickel	length	.008	.254	.002	.067	.345	.001	.256	.315	.000
	freq	-.493			-.560			-.412		
	score	.041			.047			.016		

Table 3: The coefficients, adjusted explained variance, and change in adjusted explained variance of the regression analyses on the hidden state representations learned by an MLP.

transposition and deletions play a smaller role in defining lexical neighborhoods than previously assumed.

5 Experiment 2: internal Representations

In the previous experiment, we showed that rd20 can be used to assess the neighborhood of featurized words. Calculating the rd20 over the raw features, however, assumes that our internal representations are exemplars instead of learned abstract representations, such as those found in a neural network. To assess whether rd20 can also be used with hidden state representations, we performed an additional experiment using a Multi-Layer Perceptron (MLP).

For each feature set, we trained an MLP to predict the identity of the word based on the input features, which is similar to experiments conducted by Dandurand et al. (2010). Each MLP had one hidden layer with 500 hidden units and a Sigmoid activation function, while the output layer had a softmax activation function, and a dimensionality of the vocabulary size. We used cross-entropy as a loss function, and optimized using Adam (Kingma and Ba, 2014). Our training regime was as follows: we shuffled before each epoch, and then presented all featurized words to the MLP. As in the previous experiment, we used the whole corpus for each language during training. We trained each model until convergence, where we defined convergence as there being no change in the loss for 20 epochs in a row. After convergence, we calculated the accuracy score for each of the models in each language. Each of the models achieved an accuracy of .95 or higher, showing that each model has correctly learned to predict nearly every word.

We then presented the words for which we had RTs (i.e. the words which were both in the SUBTLEX database and in the Lexicon Project for each language) to the network again, and stored the hidden unit activations in response to the input. Following the neural network literature (e.g. (Elman, 1991)), we assume these internal representations are the representations learned during the task of attempting to predict the word identity. We then calculated rd20 for each representation, and used these as input to the same analyses as the previous experiment.

Comparing the MLP results in Figure 4 to the results from Figure 3, we see that the MLP has a normalizing effect; as far as these statistics are concerned, the differences between the different feature sets have become smaller. The most prominent change is that all rd20 measures now correlate with length, whereas before only the rd20 based on slot-based values correlated with length. Similarly, the rd20 based on the one hot features did not correlate with the rd20 based on the bigram and wickelgraphs in experiment 1, but does correlate in the present experiment.

We also conducted regression analyses, using

the distances between the hidden layer representations as a predictor, as in experiment 1. Table 3 shows the results of these regression analyses. These analyses confirm that the MLP has a normalizing effect; whereas the effect of frequency and length differed in magnitude and sign between feature sets in Experiment 1, nearly all feature sets see a positive effect of length and a negative effect of frequency. The regression analysis shows that the R^2_{adj} was generally lower for the representations in the MLP, with the wickelgraphs especially suffering in comparison to Experiment 1.

6 Discussion and conclusion

Jointly, our experiments show that one hot encoded characters outperform other feature representations in explaining variance beyond frequency and length. In Experiment 1, we showed that transposition effects play a smaller role than previously thought; rd20 over a one hot encoded character representation explains significantly, albeit small amounts, more variance than old20. The rd20 of open bigrams, a feature set specifically constructed for a representation which takes into account transposition effects, does not explain a lot of variance. Returning to the main research question of this paper, i.e. whether the neighborhood effect is influenced by transposition neighbors, our evidence shows that it more likely the case that they do not.

Counter to what we found, experiments have shown that human subjects *do* take into account transposition neighbors in their neighborhoods (Davis et al., 2009; Acha and Perea, 2008). This raises an interesting conundrum, and shows that more research is required.

Furthermore, while the effect of denser neighborhoods was uniformly positive throughout all experiments and measures, this is not the case in human processing, where dense neighborhoods can sometimes have an inhibitory effect due to competition (Perea, 2015).

This leads us to another point of concern: the theoretical status of the neighborhood metric, be it old20, N, or rd20. Should these metrics be conceived of as purely diagnostic instruments, or as full-fledged, albeit limited, models of word processing? As our research shows, varying the neighborhood metric allows us to advance theoretical claims, like any model would allow us to do. In the future, we would like to investigate how much of a model one can build out of the neighborhood metric.

Experiment 2 shows the validity of using rd20 on internal representations learned by a neural network. This opens up new avenues for research, and allows us to quantitatively determine the effect of neighborhood density in neural networks on behavioral measures.

7 Implementation details

All statistical analyses were carried out using `R` (Team et al., 2013), some of the Figures were made in `ggplot2` (Wickham et al., 2008). rd20, old20 and N were implemented in `Python` (Van Rossum and Drake Jr, 1995), using `Numpy` (Walt et al., 2011), while the MLP was implemented using `PyTorch` (Paszke et al., 2017). Some Figures were made in `Matplotlib` (Hunter, 2007).

8 Acknowledgments

The first author is supported by a PhD scholarship from the FWO Research Foundation - Flanders. We would like to thank Robert Grimm and Giovanni Cassani for help with the statistical analysis and general comments. Additionally, we would like to thank the reviewers for helpful comments and suggestions, which improved the paper a lot.

References

Joana Acha and Manuel Perea. 2008. The effect of neighborhood frequency in reading: Evidence with transposed-letter neighbors. *Cognition*, 108(1):290–300.

Sally Andrews. 1997. The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. *Psychonomic Bulletin & Review*, 4(4):439–461.

R Harald Baayen, Richard Piepenbrock, and Rijn van H. 1993. The CELEX lexical data base on CD-ROM.

Marc Brysbaert and Michael J Cortese. 2011. Do the effects of subjective frequency and age of acquisition survive better word frequency norms? *Quarterly Journal of Experimental Psychology*, 64(3):545–559.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41(4):977–990.

Marc Brysbaert, Michaël Stevens, Paweł Mandera, and Emmanuel Keuleers. 2016. The impact of word prevalence on lexical decision times: Evidence from the dutch lexicon project 2. *Journal of Experimental Psychology: Human Perception and Performance*, 42(3):441.

Max Coltheart. 1977. Access to the internal lexicon. *The psychology of reading*.

Max Coltheart, Kathleen Rastle, Conrad Perry, Robyn Langdon, and Johannes Ziegler. 2001. Drc: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1):204.

Frédéric Dandurand, Jonathan Grainger, and Stéphane Dufau. 2010. Learning location-invariant orthographic representations for printed words. *Connection Science*, 22(1):25–42.

Colin J Davis. 2010a. Solar versus seriol revisited. *European Journal of Cognitive Psychology*, 22(5):695–724.

Colin J Davis. 2010b. The spatial coding model of visual word identification. *Psychological Review*, 117(3):713.

Colin J Davis and Jeffrey S Bowers. 2006. Contrasting five different theories of letter position coding: Evidence from orthographic similarity effects. *Journal of Experimental Psychology: Human Perception and Performance*, 32(3):535.

Colin J Davis, Manuel Perea, and Joana Acha. 2009. Re (de) fining the orthographic neighborhood: The role of addition and deletion neighbors in lexical decision and reading. *Journal of Experimental Psychology: Human Perception and Performance*, 35(5):1550.

Colin John Davis. 2001. *The self-organising lexical acquisition and recognition (SOLAR) model of visual word recognition.* Ph.D. thesis, ProQuest Information & Learning.

Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3):195–225.

Ludovic Ferrand, Boris New, Marc Brysbaert, Emmanuel Keuleers, Patrick Bonin, Alain Méot, Maria Augustinova, and Christophe Pallier. 2010. The french lexicon project: Lexical decision data for 38,840 french words and 38,840 pseudowords. *Behavior Research Methods*, 42(2):488–496.

Pablo Gomez, Roger Ratcliff, and Manuel Perea. 2008. The overlap model: a model of letter position coding. *Psychological review*, 115(3):577.

Jonathan Grainger. 2008. Cracking the orthographic code: An introduction. *Language and cognitive processes*, 23(1):1–35.

Jonathan Grainger and Walter JB Van Heuven. 2004. Modeling letter position coding in printed word perception.

Michael W Harm and Mark S Seidenberg. 2004. Computing the meanings of words in reading: cooperative division of labor between visual and phonological processes. *Psychological review*, 111(3):662.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

Emmanuel Keuleers, Marc Brysbaert, and Boris New. 2010a. Subtlex-nl: A new measure for dutch word frequency based on film subtitles. *Behavior research methods*, 42(3):643–650.

Emmanuel Keuleers, Kevin Diependaele, and Marc Brysbaert. 2010b. Practice effects in large-scale visual word recognition studies: A lexical decision study on 14,000 dutch mono-and disyllabic words and nonwords. *Frontiers in Psychology*, 1:174.

Emmanuel Keuleers, Paula Lacey, Kathleen Rastle, and Marc Brysbaert. 2012. The british lexicon project: Lexical decision data for 28,730 monosyllabic and disyllabic english words. *Behavior research methods*, 44(1):287–304.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sachiko Kinoshita and Dennis Norris. 2013. Letter order is not coded by open bigrams. *Journal of memory and language*, 69(2):135–150.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

James L McClelland and David E Rumelhart. 1981. An interactive activation model of context effects in letter perception: I. an account of basic findings. *Psychological review*, 88(5):375.

Risto Miikkulainen. 1997. Dyslexic and category-specific aphasic impairments in a self-organizing feature map model of the lexicon. *Brain and language*, 59(2):334–366.

Michael C Mozer. 1987. *Early parallel processing in reading: A connectionist approach.* Lawrence Erlbaum Associates, Inc.

Boris New, Marc Brysbaert, Jean Veronis, and Christophe Pallier. 2007. The use of film subtitles to estimate word frequencies. *Applied psycholinguistics*, 28(4):661–677.

Boris New, Christophe Pallier, Ludovic Ferrand, and Rafael Matos. 2001. Une base de données lexicales du français contemporain sur internet: Lexique//a lexical database for contemporary french: Lexique. *L'année psychologique*, 101(3):447–462.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Manuel Perea. 2015. Neighborhood effects in visual word recognition and reading. *The Oxford Handbook of Reading*, page 76.

Manuel Perea and Alexander Pollatsek. 1998. The effects of neighborhood frequency in reading and lexical decision. *Journal of Experimental Psychology: Human Perception and Performance*, 24(3):767.

Kathleen Rastle and Marc Brysbaert. 2006. Masked phonological priming effects in english: Are they real? do they matter? *Cognitive Psychology*, 53(2):97–145.

Jennifer Rodd, Gareth Gaskell, and William Marslen-Wilson. 2002. Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*, 46(2):245–266.

Jennifer M Rodd, M Gareth Gaskell, and William D Marslen-Wilson. 2004. Modelling the effects of semantic ambiguity in word recognition. *Cognitive Science*, 28(1):89–104.

David E Rumelhart and Patricia Siple. 1974. Process of recognizing tachistoscopically presented words. *Psychological review*, 81(2):99.

Sofie Schoonbaert and Jonathan Grainger. 2004. Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language and Cognitive Processes*, 19(3):333–367.

Mark S Seidenberg and James L McClelland. 1989. A distributed, developmental model of word recognition and naming. *Psychological review*, 96(4):523.

Mf S Seidenberg and GS Waters. 1989. Reading words aloud-a mega study.

R Core Team et al. 2013. R: A language and environment for statistical computing.

Stphan Tulkens, Dominiek Sandra, and Walter Daelemans. 2018. Wordkit: a python package for orthographic and phonological featurization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: A new and improved word frequency database for british english. *Quarterly Journal of Experimental Psychology*, 67(6):1176–1190.

Guy C Van Orden. 1987. A rows is a rose: Spelling, sound, and reading. *Memory & cognition*, 15(3):181–198.

Guido Van Rossum and Fred L Drake Jr. 1995. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. 2011. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.

Carol Whitney. 2001. How the brain encodes the order of letters in a printed word: The seriol model and selective literature review. *Psychonomic Bulletin & Review*, 8(2):221–243.

Carol Whitney, Daisy Bertrand, and Jonathan Grainger. 2012. On coding the position of letters in words. *Experimental psychology*.

Carol Whitney and Piers Cornelissen. 2008. Seriol reading. *Language and Cognitive Processes*, 23(1):143–164.

Wayne A Wickelgren. 1969. Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review*, 76(1):1.

Hadley Wickham, Winston Chang, et al. 2008. ggplot2: An implementation of the grammar of graphics. *R package version 0.7, URL: http://CRAN. R-project. org/package= ggplot2*.

Tal Yarkoni, David Balota, and Melvin Yap. 2008. Moving beyond colthearts n: A new measure of orthographic similarity. *Psychonomic Bulletin & Review*, 15(5):971–979.