Metameric

Interactive Activation at human scale

Stéphan Tulkens, Dominiek Sandra, and Walter Daelemans
CLiPS, University of Antwerp

An oft-cited shortcoming of Interactive Activation as a psychological model of word reading is that it lacks the ability to simultaneously represent words of different lengths.

We present an implementation of the Interactive Activation model, which we call Metameric, that can simulate words of different lengths, and show that there is nothing inherent to Interactive Activation which prevents it from simultaneously representing multiple word lengths. We provide an in-depth analysis of which specific factors need to be present, and show that the inclusion of three specific adjustments, all of which have been published in various models before, lead to an Interactive Activation model which is fully capable of representing words of different lengths. Finally, we show that our implementation is fully capable of representing all words between 2 and 11 letters in length from the English Lexicon Project (31, 416 words) in a single model. Our implementation is completely open source, heavily optimized, and includes both command line and graphical user interfaces, but is also agnostic to specific input data or problems. It can therefore be used to simulate a myriad of other models, e.g., models of spoken word recognition. The implementation can be accessed at www.github.com/clips/metameric.

Keywords: interactive activation, computational modeling, lexical decision

The Interactive Activation (IA) model (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982) is the oldest model of word reading, and, despite being over 30 years old, still enjoys enduring popularity in the field of computational psycholinguistics.

The model was originally proposed as an explanation of the word superiority effect (Reicher, 1969), the phenomenon that letters embedded in words are recognized more quickly than letters by themselves. To explain the word superiority effect, the model posits the existence of three separate levels of processing: a feature level, which represents the features present in letters, a letter level, which

https://doi.org/10.1075/ml.18017.tul | Published online: 14 May 2019
The Mental Lexicon 13:3 (2018), pp. 333–353. issn 1871-1340 | e-issn 1871-1375

represents abstract letter representations, and a word level, which represents whole-word orthography. Each of these three levels are connected through excitatory and inhibitory connections, where, roughly, things that are causally connected have excitatory connections. The inner workings of the model will be explained in more detail below.

McClelland & Rumelhart (1981) showed that the word superiority effect can be explained through feedback from the word to the letter level. Letters embedded in words are thus recognized more quickly because of top-down effects: words cause additional activation in their constituent letters, beyond what is expected from pure visual information.

Even though the IA model initially was launched as a model of the word superiority effect, its basic principles also lend themselves well to word recognition in general. As such, the model quickly got reinterpreted as a model of the lexical decision task. This last fact is especially interesting, as it is this reinterpretation of the IA model, a model of word recognition, and not its interpretation as an explanation of the word superiority effect, that seems to have had the most impact on the legacy of the IA model. Indeed, most models that can be considered direct descendants of the IA model are models of word recognition that do not care about the word superiority effect.

Examples of such models are the BIA (Dijkstra, Van Heuven & Grainger, 1998) and BIA+ (Dijkstra & Van Heuven, 2002) models, the DRC model (Coltheart, Rastle, Perry, Langdon & Ziegler, 2001), the MROM (Grainger & Jacobs, 1996) and MROM-P (Jacobs, Rey, Ziegler & Grainger, 1998) models, the open bigram model (Grainger & Van Heuven, 2004), SERIOL (Whitney, 2001), the spatial coding model (Davis, 2010), and the OB-1 model (Snell, van Leipsig, Grainger & Meeter, 2018). In what follows, we will refer to models that use Interactive Activation principles as "IA networks", while we will refer to the original model by McClelland & Rumelhart (1981) as the "IA model". For example, SERIOL is an "IA network" because it uses Interactive Activation principles, but it is not the same as the "IA model".

One downside of the IA model is that it uses a slot-based encoding scheme, in which letter detectors are fixed to one position, and do not affect the processing of neighboring letters. The usage of a slot-based encoding scheme has several implications. First, it implies that words are left-aligned, which means that the words "LEAD" and "PLEAD" do not share any letters. Second, it implies that the base IA model cannot account for effects of flexible letter positioning, such as the transposition effect (Grainger & Van Heuven, 2004; Grainger, 2008). For this reason, many newer IA-based models, such as the aforementioned SERIOL (Whitney, 2001) and spatial coding models (Davis, 2010) have switched to more flexible encoding schemes.

Another peculiar downside of the usage of a slot-based encoding scheme is that it restricts the IA model to words of specific lengths. The claim that IA models can only deal with words of specific lengths has been explicitly (Davis, 2003), and implicitly (Dijkstra & Rekké, 2010; Dijkstra, Wahl, Buytenhuijs, van Halem, Aljibouri, de Korte & Rekké, 2018) repeated in older and more recent work. Dijkstra et al. (2018), for example, implicitly consider the IA model length-specific, as they only use four-letter words in their IA simulations. To see why this is the case, consider the words "SWAN" and "SWANS". Clearly, when presented with the string "SWANS", the word "SWAN" would also be activated, as it shares four out of five letters with "SWANS". Nevertheless, "SWANS" would still receive more activation than "SWAN", because the final 'S' sends bottom-up activation to "SWANS", but not to "SWAN". However, it is not clear how "SWANS" and "SWAN" can be distinguished if "SWAN" is presented to the model; since there is no letter present to cause additional activation for "SWAN", both "SWAN" and "SWANS" will be activated equally.

The length-specificity is usually seen as a downside of employing IA models, as this automatically prevents many interesting interactions, such as the aforementioned interaction between "SWAN" and "SWANS". The length-specificity of the IA model is caused by the weights, i.e., the weights specified in the original IA paper (McClelland & Rumelhart, 1981), which are meant to be used with four-letter words. That is, if the original IA weights are used to simulate words of other lengths, the fit to the data will be much worse (see Experiment 2, below). Hence, even if the IA model is expanded to operate on words of lengths different from four (e.g. Grainger & Jacobs (1993) used words of length five instead of four), there is no guarantee that the weights employed in the IA model can be utilized to simultaneously model words of *different* lengths. In summary, the idea that the IA model can only deal with words of the same length, is likely caused by a mix of sociological factors, e.g., the original model only using four-letter words, and technical factors, e.g., the weights of the original model not being appropriate for words of lengths other than four.

In this paper, we show that IA networks are fully capable of simultaneously representing words of different lengths, without changing the fundamental equations governing these networks. We show that the inability to represent words of different lengths is not a consequence of the model itself, but rather of the way the data presented to the model is organized. We show that there are three specific conditions that need to be fulfilled to allow the model to represent words of different lengths. The first two conditions are both related to negative information, i.e, information about what is absent in the input. Only the addition of this information allows the model to correctly distinguish between words which are proper subsets of other words, e.g., "SWAN" and "SWANS". The third condition is

a length-dependent weight adaptation scheme. Finally, we also present Metameric, a fast simulator for IA networks that is easily extensible to cover more general localist modeling, and provide a brief introduction to its use.

The paper is structured as follows: We first introduce the IA networks and the equation governing it. Following this, we introduce three adjustments that allow the IA model to deal with words of different lengths. We then perform three experiments: first, we reproduce the results of the original IA model using four-letter words (McClelland & Rumelhart, 1981), thereby validating our implementation. In the second experiment, we introduce all three adjustments separately, and measure the impact each of these adjustments has on the correlation between the model output and Reaction Times (RT). In the third experiment, we model the entire English Lexicon Project (Balota, Yap, Hutchison, Cortese, Kessler, Loftis, Neely, Nelson, Simpson & Treiman, 2007) in a single model, and show that this increases the correspondence between cycle times and Reaction Times, showing that the extending the IA model in this manner is useful for future experiments. Finally, we explain the standard usage of Metameric with regard to experiments involving IA networks.

Interactive Activation networks: The general framework

To make the adjustments to the model as clear as possible, we will first outline the general framework of Interactive Activation.

An Interactive Activation network is a localist connectionist model, in which nodes are organized into layers, which are connected by excitatory or inhibitory links. An IA network is called localist because each of its neurons, or nodes, uniquely identifies some item (Page, 2000). Localist representation contrasts with distributed representation, in which items are uniquely identified by the joint activation pattern over multiple nodes. In the framework of IA, layers represent groupings of items, and nodes belong to a single layer. For example, the nodes in the IA model are organized into a feature layer, a letter layer, and a word layer, in which nodes uniquely represent single features, single letters, and single word, respectively.

Every node has inhibitory or excitatory connections to other nodes. Importantly, while connections between layers can be either inhibitory or excitatory, the connections within a layer are always inhibitory.

Besides inhibitory and excitatory connections, each node also has an associated activation level, which denotes the degree of activation of that node, and a resting level activation (RLA), which denotes the activation level to which the node returns in absence of external input. In general, IA models have excitatory

connections between nodes whose presence implies the other, inhibitory connections between nodes whose co-existence is contradictory, and no connections between nodes that do not co-occur systematically. For example, the node that represents the letter "A" in the first position has excitatory connections to words like "ALCOVE" and "ARMOUR". This same "A" also has inhibitory links to all other words in the lexicon that do not have an "A" as their first letter. Similarly, the letter "A" has no links to nodes in the second slot, as the presence of the letter "A" in the first slot neither confirms nor denies the existence of any other letter in the second slot.

Data is presented to an IA network by "clamping" nodes in one or more layers, which sets them to a given value, and then updating the model until some threshold for recognition is met. The number of updates it takes for the model to converge is commonly known as the cycle time, and is assumed to be a direct analogue to reaction time in behavioral experiments (Grainger & Jacobs, 1996; Jacobs et al., 1998; Davis, 2003; Davis & Lupker, 2006).

The update equations for the IA model are as follows:

First, for a given node with index *i*, the net input is calculated. The net input is the sum of all active nodes multiplied by their weights, where an active node is a node with an activation > 0.

$$net_i = \sum_j \begin{cases} w_{ij} a_j, & a_j > 0 \\ 0, & a_j \leq 0 \end{cases}$$

Where w_{ij} is the weight matrix between nodes. For notational simplicity, we assume that nodes that are not connected have a weight of 0. Similarly, excitatory and inhibitory connections positive and negative weights, respectively. Given the net input, the update of the node ext_i is calculated as follows:

$$ext_i = \begin{cases} (\max - a_i) net_i, & net_i > 0 \\ (a_i - \min) net_i, & net_i \leq 0 \end{cases}$$

Where a_i is the activation of the node, *max* is the maximum activation, and *min* is the minimum activation. In all our simulations *max* and *min* are set to 1.0 and −0.2, respectively. This update equation ensures that the update gets smaller as the activation of the unit in question gets closer to the maximum or minimum values.

In tandem with ext_i, the decay, d_i is calculated:

$$d_i = decay(a_i - rest_i)$$

Decay is responsible for driving the activation back to the resting level, the magnitude of which is dependent on the difference between the activation and the

resting level; the further away the node currently is from the resting level, the larger the decay factor is. This weight decay is scaled by a weight, *decay*, which we set to .07. The decay and external input are then combined, as follows:

$$a_i^{t+1} = a_i^t + ext_i - d_i$$

This update rule clearly demonstrates the two competing forces at work in IA networks; each node is driven by external input, but simultaneously desires to return to its resting level activation. It is worth noting that IA networks are member of a larger class of models, often called resonance theory models (Grossberg, 1978), although they are not necessarily completely compatible with them (Grossberg, 1987).

The IA model

An IA model is an IA network which consists of three separate layers. The bottommost layer is the feature layer, which contains position-specific letter features in a fourteen segment encoding (Rumelhart & Siple, 1974). The fourteen segment encoding broadly encodes the visual similarity between letters by assigning line segments in letters separate features.

The feature layer links to a letter layer, in which each letter in each position is assigned a separate node, e.g., the letter "G" in the first position is both different from "S" in the first position and "G" in the second position. Each feature in the feature layer has an excitatory connection to the letters which contain that feature, and inhibitory connections to letters that do not contain that feature. Finally, the letter layer links to a word layer, in which nodes represent whole (orthographic) words. The word and letter layers are related like the letter and feature layers; words that contain a letter in a specific position have an excitatory connection from that specific letter node, and inhibitory connection to all other letter nodes in that position. The word nodes also have a feedback connection to the letter nodes; words containing a specific letter excite the node assigned to that specific letter.

The three adjustments

In this section, we will explain the three adjustments we made to the model.

Negative features

The first adjustment concerns the feature level. As explained above, the nodes of the feature level correspond to line segments in letters, where features have

excitatory connections to letters that contain them, and inhibitory connections to letters that do not contain them. While this situation seems to cover the situations that apply to our understanding of the world; features are either present or absent, it leads to a situation in which negative evidence is not correctly utilized by the model.

As an example, consider that all features having an excitatory connection to "O" also have excitatory connections to "Q", that is, the features with excitatory connections to "O" are a proper subset of those with excitatory connections to "Q" Recall that, according to equation 1, only active nodes contribute to activation. This implies that the presentation of the features contained in the letter "O" leads to the equal activation of the letter "Q", as all features linked to "O" are also linked to "Q". The IA model effectively does not have a way of representing the absence of a given feature; a feature which is set to 0, and effectively absent, does not inhibit or excite anything. The sole feature that is not shared by "Q" and "O", the diagonal line segment, is not present when "O" is present, and hence does not affect the activation of the model.

The only way to solve this in the framework of IA is to add the absent features as having excitatory connections to letters that do not contain them. Adding these negative features is equivalent to a procedure known as complement coding (Carpenter, Grossberg & Rosen, 1991b), and is utilized heavily in Adaptive Resonance Theory (Carpenter and Grossberg, 1987b) models, such as ART (Carpenter & Grossberg, 1987a) and ARTMAP (Carpenter, Grossberg & Reynolds, 1991a). Complement coding consists of normalizing all features so that they fall into the interval (0,1), and then adding a negative feature x' for each positive feature x which has the value $1-x$. For binary features, which are by definition in (0,1), complement coding simply consists of adding a feature value of 1 for each value of 0 and vice versa.

After adding the negative features, the letters "O" and "Q" still largely overlap in their features, but now the model has the power to differentiate between "O" and "Q", as the absence of the small diagonal line segment in the "Q" negatively contributes to the activation of "Q", but positively contributes to the activation of "O".

Note that it is unclear whether this adjustment was present in the original IA model. For one, the appendix of the original paper introducing IA mentions the following: (McClelland & Rumelhart, 1981):

> There is another array that holds the information the models has detected about the display. Each element of this array represents a detector for the presence or absence of a feature. When the corresponding feature is detected, the detector's value is set to 1. (remember that both absence and presence must be detected)

We therefore entertain the possibility that the negative features were already present in the original formulation of IA, and test this possibility specifically in Experiment 1.

Space padding

As we saw in the previous section, negative features are necessary to be able to correctly distinguish between letters whose features are a proper subset of the features of a different letter in the same position. A similar situation applies to words, although it can only arise if words with different lengths are considered within the same model.

As an example, consider the words "SWAN" and "SWANS". Like "O" and "Q", the letters contained in "SWAN" are a subset of those in "SWANS". Hence, if the word "SWAN" is presented to the model, the word representations of "SWAN" and "SWANS" are both activated equally. Moreover, due to the feedback from "SWANS" to the letter "S" in the fifth position, "SWANS" ends up accumulating more activation, and ends up being the word with the highest activation, which is an undesirable situation.

This problem can be solved by adding explicit space characters as padding to each word, and by considering these space characters as full letters in the letter layers. These space characters thus have inhibitory links to all features signaling the presence of a line segment, and excitatory links to all features signaling the absence of a line segment.

If a space character is added explicitly, the presentation of "SWAN" to the model will no longer activate "SWAN" and "SWANS" equally, because "SWANS" will be inhibited by the space character, and "SWAN" will be activated by the space character.

Note that, unlike the addition of negative features, this is not equivalent to complement coding, as there is only one negative feature for each slot.

This innovation was already included in the implementation of the DRC model of Coltheart et al. (2001).

Weight adaptation

Finally, we also include an adjustment we call weight adaptation, in which the weights of the IA model are adapted to the length of the words in the model. This innovation was first proposed and used by Loncke, Martensen, Van Heuven & Sandra (2009). To adapt the weights of the IA model to a specific length, we multiply the weights between the word and letter layers of the original IA model by four (i.e., the number of slots of the original IA model.) Then, given a lexicon

with words of variable length, we divide the excitatory connections by the length of the longest word in the lexicon, while multiplying the inhibitory connections by the length of the longest word in the lexicon.

As an example, consider the excitatory connections between words and letters, which are set to .3 in the original formulation of the IA model (McClelland & Rumelhart, 1981). In our model we first multiply this weight by 4, and then divide this by the number of desired slots in the simulation. If the longest word in our simulation is, for example, 7 letters, our final weight would be $\frac{(.3 \times 4)}{7} = .171$.

Because all words are padded with space characters, this ensures that the amount of excitatory and inhibitory information flowing between words is kept constant.

Experiments

To show the influence of each of the proposed innovations, we ran multiple experiments. In all experiments, we used two measures to quantify how well our models are doing: accuracy and correlation to lexical decision Reaction Times (RT).

Accuracy is defined as the proportion of items which were correctly classified:

$$accuracy(correct, incorrect) = \frac{|correct|}{|correct| + |incorrect|}$$

Where correct and incorrect are sets containing the instances where the model was correct and incorrect, respectively. The |.| operator is the cardinality operator.

These measures are warranted because we require that a good model of word reading simultaneously produces the correct response to a word, while also correlating well with behavioral measures. We use Lexical Decision RT measurements as our behavioral measure because the IA model implements a theory of lexical decision, and therefore ought to correlate with RT measurements.

We measure accuracy by storing the most active word after convergence on presentation of a word. If the orthographic form of the most active word after convergence was equal to the presented word, we counted it as correct. Following Grainger & Jacobs (1996), if the model did not converge, i.e., it did not reach the threshold of .7 before a pre-specified large number of cycles was reached, we counted it as incorrect. In all experiments, we only calculated the correlation between cycle times for words which were correctly recognized.

Data

All experiments were carried out using a subset of the English Lexicon Project (ELP) (Balota et al., 2007). Specifically, we extracted all words from the ELP between a length of 3 and 10 inclusive. We removed any words which did not solely consist of alphabetic characters, e.g., words containing the genitive marker, dashes, or spaces. This remaining subset consisted of 29,173 words.

Experiment 1. Replication

In this experiment, we run the original simulations of the IA model using all four-letter words in our lexicon, a set of 1907 words. Out of this set of words, we sampled 100 subsets of size 1430 (75% of 1907), based on the log frequency of those words. Sampling allows us to effectively simulate multiple participants, and simulate the effect of different corpora on the performance of the IA model. Our sampling strategy was as follows: we first binned the original data into bins of width 1, based on their log frequency, i.e., all words whose log frequency started with the same number were put in the same bin. We then noted the proportion of the number of items in each bin, compared to the total number of items. We then sampled a number of items from each bin, proportional to their total number of items in the corpus. This ensures that even low-frequency items get sampled often enough for us to have reliable estimates, and ensures that high-frequency items do not dominate our results.

When analyzing the results, we saw that the original IA model does not achieve an accuracy of 1.0, as seen in Figure 1. The main cause for this seems to be the lack of negative feature input. We see, for example, that "ZING" and "ZINC" are activated equally on presentation of "ZINC", thereby keeping the model from converging. This is because the features in the letter "G", as mentioned above, are a strict superset of the features in the letter "C".

This makes it likely that the original IA model does include negative features, as hypothesized above. We therefore run the same simulation again, this time including the negative features, which brings it in line with the expected behavior of the model.

Figure 1 shows the distribution over accuracy scores of both models. The model without negative features achieves an accuracy of 0.756 (+/− 0.01), while the model incorporating negative features achieves an accuracy of 1.0 (+/− .0). This clearly shows that the addition of negative features is required for the model to function.

Figure 2 shows the correlation scores between cycle times and RT measurements from the ELP. The model without negative features does not correlate well

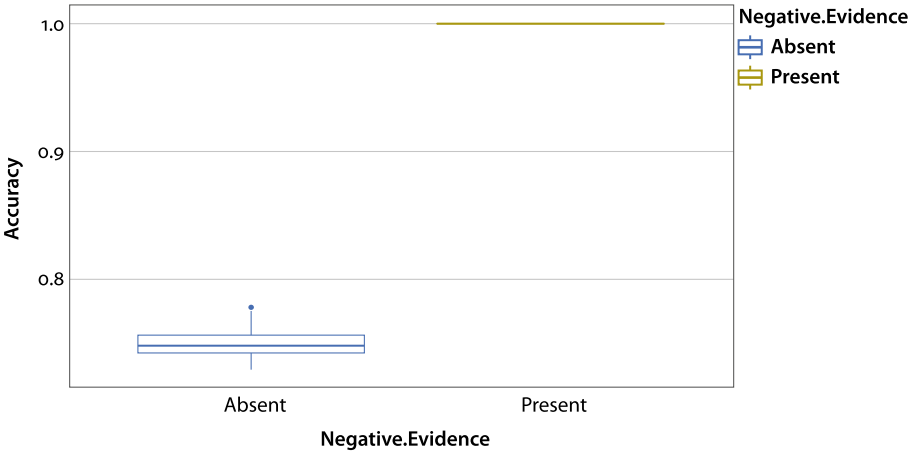

Figure 1. The accuracy scores of both models in Experiment 1. This clearly shows that the model requires negative features to function correctly

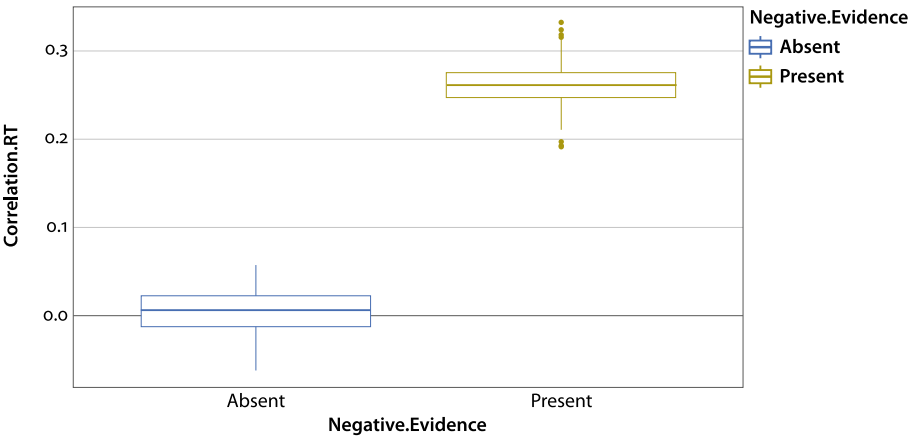

Figure 2. The correlations with RT of both models in Experiment 1. This also shows shows that the model requires negative features to function correctly

with RT measurements, and achieves a mean correlation of −0.026 (+/− 0.032). The model that includes negative features achieves mean correlation of 0.26 (+/− 0.02), indicating a fair correlation with RT. Again, we see an enormous influence of the inclusion of negative features.

Experiment 2. Adding the adjustments

In this set of experiments, we ran models on sampled subsets of the 29,173 words selected from the ELP. Because we have three adjustments, we can define eight pairwise combinations of these three adjustments, leading to a set of eight model types.

As in the previous experiment, we sampled 100 subsets by their log frequency, although the size of these subsets was 25% of the full set (7293 words) instead of 75%. We lowered the number of items in each sample because of computational reasons; our IA simulator is fast, but IA in general slows down immensely for degenerate cases in which the threshold is not reached.

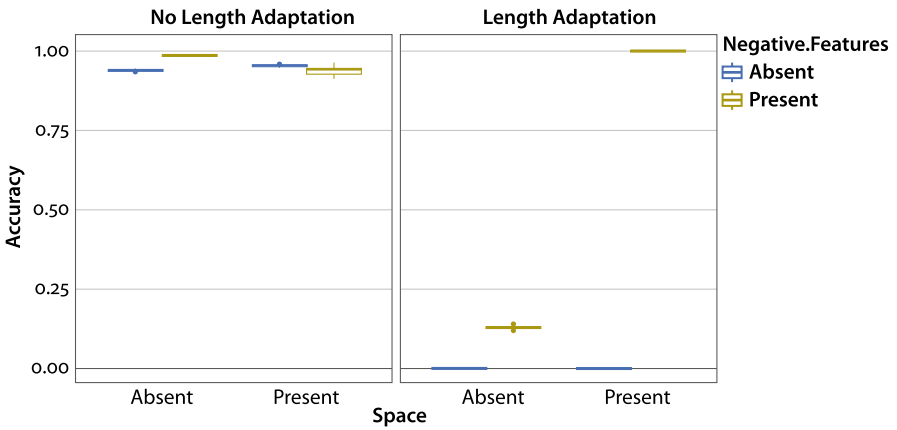

Figure 3. The accuracy scores of the models using the various adjustments. As the figure shows, only the model using all three adjustments achieves the desired accuracy score of 1.0. The model without any adjustments still achieves a high accuracy score. Moreover, the addition of length adjustment without the addition of the other adjustments has a negative effect on performance

The accuracy scores of the various models are shown in Figure 3. As before, the distributions are calculated over the 100 subsets. The only model achieving an accuracy of 1.0, which is desirable for models of word reading, is the model with all three adjustments. The accuracy of models without length adjustment is also uniformly high, and ranges between .93 and .98, although it never reaches 1.0. Models with length adjustment that do not have both other adjustments have uniformly low scores.

That models without adjustments, or models with only some adjustment, also get high accuracy scores raises the question of whether the adjustments are useful at all.

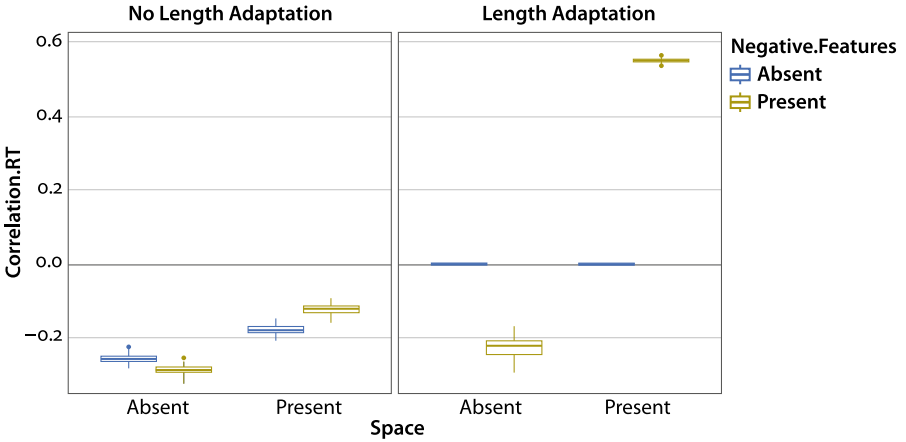

Figure 4. Correlations with RT of the models with various adjustments. As this graph shows, there is a uniformly negative trend for all models that do not include all positive features

Considering the correlation with RT, however, we see that only the model with all three adjustments has a positive correlation with RT. Figure 4 shows the correlations between RTs and cycle times. While the models without the adjustments have high categorization accuracy, their correlation with RT measurements is negative, which is the opposite of what we expect.

The cycle time of the model with all three adjustments correlated well with RT 0.54 (+/− 0.005). Note that this correlation is also higher than the 0.26 reported for the model with only four-letter words. This underscores that it is important to create models which can account for the full range of empirical data before drawing conclusions from them; adding words of different lengths to the model apparently causes the model to more closely fit empirical data.

Finally, we also expect cycle time to vary with word length; a situation which, until now, was not testable within the framework of Interactive Activation. The results of this analysis are shown in Figure 5.

This, again, shows that the adjustments are necessary; models without all adjustments have negative correlations with word length, which is contrary to empirical evidence.

Experiment 3. Simulating the English Lexicon Project

Finally, we perform a single experiment in which we simulate all 31,416 words selected from the ELP in a single model. For this model, the correlation between cycles and RT was 0.57, showing results consistent with Experiment 2, and which confirms our hypothesis that adding more words leads to a better model fit.

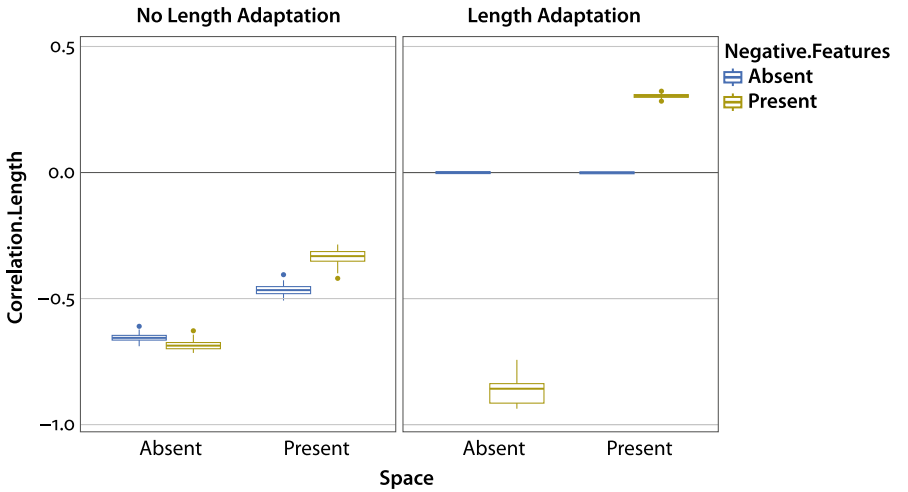

Figure 5. The correlation of various models with length

We also plot the cycles versus RTs for all words, together with a single linear regression with cycles as independent variable and RT as a dependent variable. This is shown in Figure 6, and shows that the IA model tends to overestimate the cycle times for words with lower RTs.

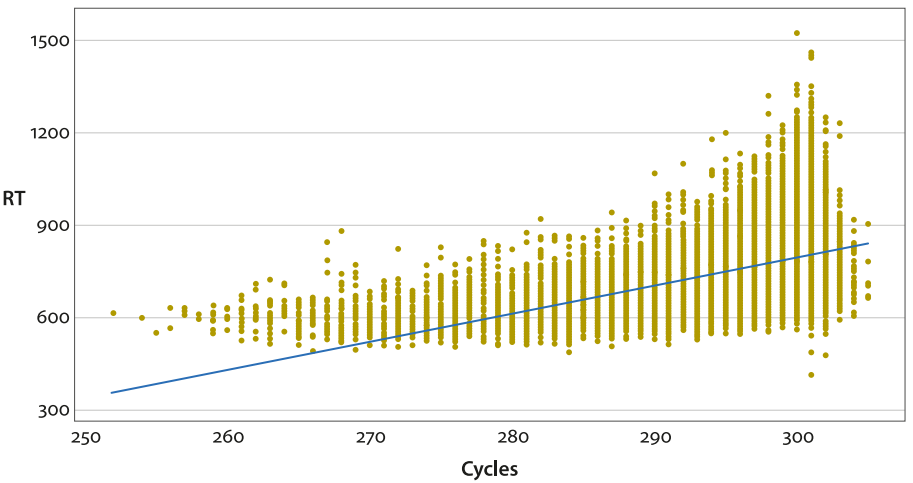

Figure 6. The scatter plot of Experiment 3. The y-axis is the RT and the x-axis is the cycle time. The red line shows a simple linear regression with cycle time as an independent variable and RT as the independent variable

Experiment 4. Comparison with jIAM

The only other currently available Interactive Activation simulator is jIAM, which is only available as a web interface, and closed source. As a comparison to jIAM, we presented the corpus used in experiment 1, i.e., the 1907 four-letter words from the ELP, to the web interface of jIAM and Metameric.

jIAM took 15 minutes and 12 seconds to process the words, while Metameric took approximately 3 seconds. Additionally, we attempted to run jIAM on the full ELP, but it crashed after presentation of the first word.

A caveat given these timings: jIAM is only available as a web application, and therefore necessarily involves the rendering of an interface, which takes processing time.

Nevertheless, these timings still give a good indication of the time a researcher or user has to spend waiting for results, and show that our implementation is able to represent vastly more words than jIAM.

Intermediary conclusion: A common thread

In the previous section, we showed that the only way to guarantee correspondence with empirical results is to implement exactly these three adjustments. What is lacking so far is a rationale as to why these three adjustments are all necessary for the model to function. We argue that the single principle behind all three adjustments is normalization: all three adjustments ensure that the amount of activation flowing into, and out of, each node is constant. Without negative features, for example, words with more features would get more activation. After adding negative features, each node has 14 excitatory connections to 14 different features, some of which are positive, and some of which are negative. Similarly, before adding the space character, longer words have more incoming activation, simply because these words have more characters.

Hence, the addition of space characters and negative features both cause normalization because they ensure that words and letters have the same number of incoming connections, while weight adaptation normalize the total positive and negative activation flowing into nodes not associated with a length.

Metameric

In addition to the theoretical arguments above, we also present a fast IA simulator, called Metameric. We will briefly describe the typical overview of a Metameric experiment, although we will refrain from describing the inner workings of the simulator, as these will be subject to change in the future.

An experiment with Metameric generally encompasses two stages. First, data needs to be prepared, which means that the appropriate letter features and letter segment features are assigned to each word. Second, the previously prepared data is fed into the model, which can then be used to run an experiment. Before describing these two steps, we will describe the kind of input data Metameric accepts.

Data input

In order to run an experiment, Metameric must be provided with two separate input files: and Comma Separated Value (CSV) file containing the weights, which we call the weight file, and a second CSV files containing the data, which we call the lexicon.

The weight file contains the positive and negative connections between each layer in the model to be created, and hence also specifies the structure of the layers in the IA model.

The lexicon, on the other hand, specifies which nodes will be added to each layer: each row in the lexicon is assumed to be a separate item, while each column is assumed to correspond to characteristic which can be used in the creation of the IA model. Note that only columns that are present in the weight file will be used in the IA model. This has the advantage of being able to directly use well-known databases, such as those from the ELP (Balota et al., 2007) and other Lexicon Projects without having to sanitize or otherwise clean them.

Because of this structure, Metameric is a completely data-driven IA simulator; the user does not have to prespecify features, nodes or symbols in advance, there are all derived from the input data, which ensures both a parsimonious as well as a consistent model.

Preparing the lexicon

As an IA network is a localist connectionist model, the main effort in creating a model is setting up the lexicon in an appropriate format. To facilitate this, we include a preparation utility that can be used to decompose and featurize fields. Decomposition is defined as turning an otherwise symbolic string of letters, phonemes, or other characteristics into a slot-based set of items. For example, the decomposition of the string "DOG" into letters is "D-0 O-1 G-2". Of course, not all fields are amenable to decomposition, an example of this being a system which contains semantic symbols.

Featurization is carried out on a slot-based decomposed layer, e.g. the letter layers as above, and is defined as turning each of the individual items in each slot

into a number of features, as in the canonical IA model. An important note is that featurization can be carried out on fields which are created by the decomposition process.

As Metameric is completely agnostic to its input data, it is a generic IA network simulator, models with phonological layers, such as the MROM-p (Jacobs et al., 1998) model, or language nodes, such as the BIA (Dijkstra et al., 1998) and BIA+ (Dijkstra & Van Heuven, 2002) can also be easily constructed within Metameric.

Running an experiment

After preparing the lexicon, it can be used to construct and run a model. As explained above, the model is automatically constructed upon being fed a lexicon and a weight matrix. Both the web interface and the CLI offer a great deal of control over all parameters of the IA model. As said above, all other weights and connections in the IA model are automatically set based on the data in the input lexicon.

After creating a model, it can be fed data from the lexicon, or entirely new data. Data which is fed into the model is returned in exactly the same format, i.e., a CSV with item characteristics, but with an added column of cycle times.

Extending metameric

Metameric is easily extensible, and can be directly used within python scripts to provide a great deal of control over the simulator. The core components of Metameric are heavily optimized, and completely agnostic to the input data, and can therefore be used to simulate different Interactive Activation networks.

The web interface

The web interface offers a concise and user-friendly way for non-programmers to automatically construct IA models and run experiments using their own stimuli. After starting the web interface, the user simply has to visit a locally hosted website, which will then show the Metameric web interface.

The website contains the following components, which are grouped under separate tabs.

Experiment

Perform an experiment using formatted CSV data and user-defined parameters. The Experiment screen is shown in Figure 7.

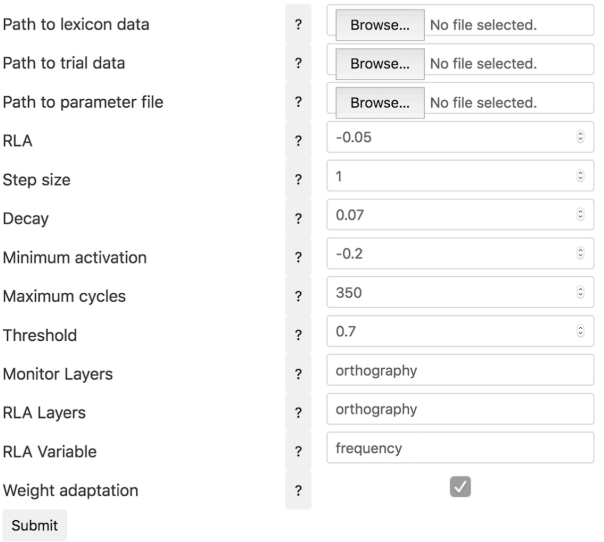

Figure 7. The experiment screen with the default parameters

Prepare

The prepare tab is used to prepare data for input in an IA model, and provides a concise interface to the preparation module detailed above. The prepared CSVs are stored locally, allowing users to freely switch between using the web interface and CLI after preparing the data.

Analyze

Finally, the web interface allows users to interactively plot model responses. After model creation, the user is led to a screen in which input can be interactively provided to the model, allowing the user to inspect how the model reacts to various words or non-words. An example of the analysis screen is shown in Figure 8.

Conclusion

In this paper, we have presented an analysis of one of the main shortcomings of the Interactive Activation model; the inability of the model to deal with words of different lengths. We have shown that there is nothing inherent to the IA model that prevents it from modeling words of different lengths. Although slot-based encodings clearly have downsides in that they do represent letter order explicitly,

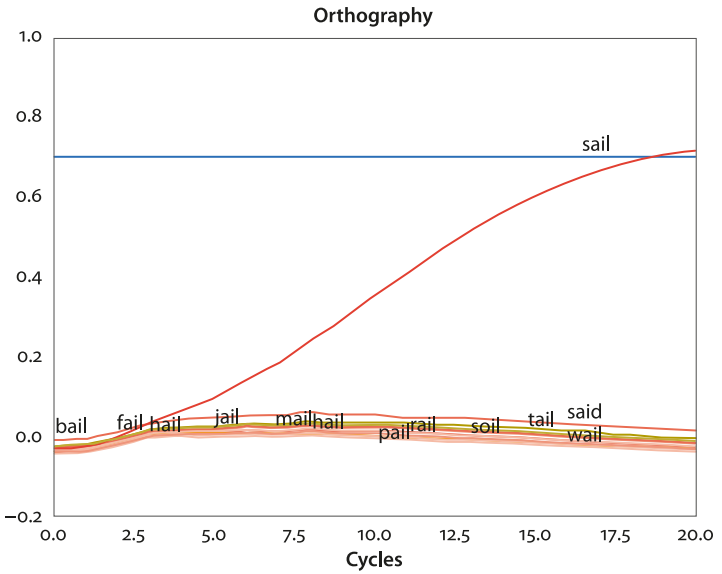

Figure 8. An example of an analysis picture produced by Metameric

it is important that we know the limits of these encodings and models, both for our theoretical understanding of the IA model, and localist modeling in general.

In addition to the analyses, we also presented a fast and highly extensible IA simulator, called Metameric, which is open source and highly extensible. We hope that both parts of this paper, i.e., the analysis and the simulator, will lead to future research into Interactive Activation models, and Interactive Activation networks in general.

References

Balota, D. A., Yap, M. J., Hutchison, K. A., Cortese, M. J., Kessler, B., Loftis, B., Neely, J. H., Nelson, D. L., Simpson, G. B., and Treiman, R. (2007). The English lexicon project. *Behavior research methods*, 39(3), 445–459. https://doi.org/10.3758/BF03193014

Carpenter, G. A. and Grossberg, S. (1987a). Art 2: Self-organization of stable category recognition codes for analog input patterns. *Applied optics*, 26(23), 4919–4930. https://doi.org/10.1364/AO.26.004919

Carpenter, G. A. and Grossberg, S. (1987b). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1), 54–115. https://doi.org/10.1016/S0734-189X(87)80014-2

Carpenter, G. A., Grossberg, S., and Reynolds, J. H. (1991a). Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural networks*, 4(5), 565–588. https://doi.org/10.1016/0893-6080(91)90012-T

Carpenter, G. A., Grossberg, S., and Rosen, D. B. (1991b). Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural networks*, 4(6), 759–771. https://doi.org/10.1016/0893-6080(91)90056-B

Coltheart, M., Rastle, K., Perry, C., Langdon, R., and Ziegler, J. (2001). DRC: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1), 204. https://doi.org/10.1037/0033-295X.108.1.204

Davis, C. J. (2003). Factors underlying masked priming effects in competitive network models of visual word recognition. *Masked priming: The state of the art*, 121–170.

Davis, C. J. (2010). The spatial coding model of visual word identification. *Psychological review*, 117(3), 713. https://doi.org/10.1037/a0019738

Davis, C. J. and Lupker, S. J. (2006). Masked inhibitory priming in english: Evidence for lexical inhibition. *Journal of Experimental Psychology: Human Perception and Performance*, 32(3), 668.

Dijkstra, T. and Rekké, S. (2010). Towards a localist-connectionist model of word translation. *The Mental Lexicon*, 5(3), 401–420. https://doi.org/10.1075/ml.5.3.08dij

Dijkstra, T. and Van Heuven, W. J. (2002). The architecture of the bilingual word recognition system: From identification to decision. *Bilingualism: Language and cognition*, 5(3), 175–197. https://doi.org/10.1017/S1366728902003012

Dijkstra, T., Van Heuven, W. J., and Grainger, J. (1998). Simulating cross-language competition with the bilingual interactive activation model. *Psychologica Belgica*.

Dijkstra, T., Wahl, A., Buytenhuijs, F., van Halem, N., Al-jibouri, Z., de Korte, M., and Rekké, S. (2018). Multilink: A computational model for bilingual word recognition and word translation. *Bilingualism: Language and Cognition*, 1–23.

Grainger, J. (2008). Cracking the orthographic code: An introduction. *Language and cognitive processes*, 23(1), 1–35. https://doi.org/10.1080/01690960701578013

Grainger, J. and Jacobs, A. M. (1993). Masked partial-word priming in visual word recognition: Effects of positional letter frequency. *Journal of experimental psychology: human perception and performance*, 19(5), 951.

Grainger, J. and Jacobs, A. M. (1996). Orthographic processing in visual word recognition: A multiple read-out model. *Psychological review*, 103(3), 518. https://doi.org/10.1037/0033-295X.103.3.518

Grainger, J. and Van Heuven, W. J. (2004). Modeling letter position coding in printed word perception.

Grossberg, S. (1978). A theory of visual coding, memory, and development. *Formal theories of visual perception*, 7–26.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1), 23–63. https://doi.org/10.1111/j.1551-6708.1987.tb00862.x

Jacobs, A. M., Rey, A., Ziegler, J. C., and Grainger, J. (1998). Mrom-p: An interactive activation, multiple readout model of orthographic and phonological processes in visual word recognition. In Grainger, J. and Jacobs, A., editors, *Localist connectionist approaches to human cognition*, 147–188. Lawrence Erlbaum Associates Publishers.

Loncke, M., Martensen, H., van Heuven, W. J., and Sandra, D. (2009). Who is dominating the dutch neighbourhood? On the role of subsyllabic units in dutch nonword reading. *The Quarterly Journal of Experimental Psychology*, 62(1), 140–154. https://doi.org/10.1080/17470210701851206

McClelland, J. L. and Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological review*, 88(5), 375. https://doi.org/10.1037/0033-295X.88.5.375

Page, M. (2000). Connectionist modelling in psychology: A localist manifesto. *Behavioral and Brain Sciences*, 23(4), 443–467. https://doi.org/10.1017/S0140525X00003356

Reicher, G. M. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of experimental psychology*, 81(2), 275. https://doi.org/10.1037/h0027768

Rumelhart, D. E. and McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: II. The contextual enhancement effect and some tests and extensions of the model. *Psychological review*, 89(1), 60. https://doi.org/10.1037/0033-295X.89.1.60

Rumelhart, D. E. and Siple, P. (1974). Process of recognizing tachistoscopically presented words. *Psychological review*, 81(2), 99. https://doi.org/10.1037/h0036117

Snell, J., van Leipsig, S., Grainger, J., & Meeter, M. (2018). OB1-reader: A model of word recognition and eye movements in text reading. *Psychological review*, 125(6), 969. https://doi.org/10.1037/rev0000119

Whitney, C. (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. https://doi.org/10.3758/BF03196158

Address for correspondence

Stéphan Tulkens
CLiPS, University of Antwerp
Prinsstraat 13
2000, Antwerpen
Belgium
stephan.tulkens@uantwerpen.be

Co-author information

Dominiek Sandra
CLiPS
University of Antwerp
dominiek.sandra@uantwerpen.be

Walter Daelemans
CLiPS
University of Antwerp
walter.daelemans@uantwerpen.be