Biodiversity in NLP: modelling lexical meaning with the Fruit Fly Algorithm

Simon Preissner*
University of Trento

Aurélie Herbelot**
University of Trento

The natural world is very diverse in terms of biological organisation, and solves problems in a wide variety of efficient and creative manners. This biodiversity is in stark contrast with the landscape of artificial models in the field of Natural Language Processing (NLP). In the last years, NLP algorithms have clustered around a few very expensive architectures, the cost of which has many facets, including training times, storage, replicability, interpretability, equality of access to experimental paradigms, and even environmental impact. Inspired by the biodiversity of the real world, we argue for a methodology which promotes 'artificial diversity', and we further propose that cognitively-inspired algorithms are a good starting point to explore new architectures. As a case study, we investigate the fruit fly's olfactory system as a distributional semantics model. We show that, even in its rawest form, it provides many of the features that we might require from a good model of meaning acquisition, and that the original architecture can serve as a basis for cognitively-inspired extensions. We focus on one such extension by implementing a mechanism of neural adaptation.

1. Introduction

The natural world is diverse. Biological species exhibit a huge variety of genetic make-ups, and by extension, a wealth of different morphologies, functions and behaviours. In comparison, there is little diversity in computational models of Natural Language Processing (NLP). In this paper, we will argue that this lack of heterogeneity is detrimental to finding solutions to core problems, especially when dominant paradigms fail to satisfy linguistic, cognitive, and/or ethical requirements (Linzen 2020; Bender and Koller 2020; Hovy and Spruit 2016; Köhl et al. 2019). We will also offer an alternative experimental paradigm, inspired by the variegated nature of the real world.

Let us start by noting that in recent years, the NLP community has seen an increase in the popularity of expensive models requiring enormous computational resources to train and run. The cost of such models is multi-faceted. From the point of view of shaping the scientific community, they create a huge gap between researchers in wealthy institutions and those with less resources and they often make replication prohibitive. From the point of view of applicability, they make the end-user dependent on high-tech hardware which they may not afford, or on cloud services which may have problematic privacy side-effects (and are not available to those with poor Internet access). Training such models can often take a long time and extraordinary amounts of energy, generating

* Centro Interdipartimentale di Mente e Cervello (CIMeC) - Corso Bettini 31, 38068 Rovereto (TN), Italy - E-mail: simon.preissner@gmx.de

** Center for Mind/Brain Sciences (CIMeC) - Corso Bettini 31, 38068 Rovereto (TN), Italy; Department of Information Engineering and Computer Science (DISI) - Via Sommarive 9, 38123 Povo (TN), Italy. E-mail: aurelie.herbelot@unitn.it

CO_2 emissions disproportionate to the models' improvements (Strubell, Ganesh, and McCallum 2019). From a pure modelling point of view, finally, complexity often comes with a loss of interpretability, which weakens theoretical insights. Whilst we appreciate that a part of NLP is focused on engineering applications rather than simulating natural language proper, it seems that the community would benefit from a more comparative approach to modelling, and from a diversification of algorithms.

By analogy to bio-diversity, we will therefore argue for a notion of 'artificial diversity' and introduce an experimental paradigm that would foster such heterogeneity of models. We will further contend that a good place to find smaller and more interpretable algorithms is indeed the natural world. Beyond the actual human brain, known as an extremely efficient learner and storage system, many organisms display core cognitive abilities such as incremental learning, generalisation or classification, which many NLP systems need to emulate. Such faculties develop in extremely simple systems, which are good contenders for the type of models we advocate here. Investigating those, however, requires a clear stance on evaluation: we cannot expect a very simple model to beat the performance of heavily-trained systems, but we can require it to give satisfactory results whilst also being a good *model* in the strong sense of the term, that is, simulating all observable features of a given real-world phenomenon. Thus, we propose a methodology focused on the identification of general modelling requirements, which we imagine being applied to a wide array of algorithms for comparison, not competition.

This paper is an extension of our original work on the Fruit Fly Algorithm (FFA), showing that the olfactory system of the fruit fly can be be used to learn word embeddings with little complexity and added transparency (Preissner and Herbelot 2019). In addition to the original material (§4 and §5.1), we clarify our methodological claim (§2), emphasising the steps that we feel are important when designing an NLP system with diversity in mind. We also include an extension of the original FFA, modelling the natural neural adaptation process in living organisms, i.e. the decrease in response to a frequent stimulus (§5.2). We show the capabilities of the modified FFA on a word vector learning task, illustrating its fully incremental behaviour and assessing its level of interpretability with respect to other word embedding methods (§7).

2. A methodology for artificial diversity

The standard experimental paradigm in NLP consists in setting up a task which models can compete over, with the view of getting the best possible performance on that task. This practice has the disadvantage of focusing efforts on 'the most promising' models, from the point of view of performance, regardless of cost and characteristics. As pointed out by (Linzen 2020), this favours models trained on huge amounts of data, with little cognitive plausibility and little human-like generalisation power.

We propose instead to encourage paradigmatic diversity by evaluating a model in terms of a set of theoretical requirements, *as well as* performance. For instance a model may provide state-of-the-art results on a task while failing at incrementality. Another one may implement incrementality but fall short at learning from small data. We suggest that an analysis of very different architectures may be more beneficial to our understanding of human language than 'solving' a task in raw terms (getting the best score).

With this in mind, the methodology we propose rests on a careful analysis of requirements, with respect to a task or phenomenon. We suggest a pipeline in five steps:

Description of the problem: a given task or phenomenon can be described in different ways depending on the end goal of the computational simulation. A model produced with applications in mind might explain how the task relates to a real-world need. A model produced for fundamental research might give a description of a phenomenon from a particular perspective (linguistic, cognitive, biological, etc).

Identification of requirements: in addition to quantifiable performance, a model should satisfy a number of architectural desiderata. For applications, work might focus on performance itself, especially for tasks where accuracy is critical (e.g. decision-making tools, medical applications for end-users). We would also expect some reflection on the ethical implications of the task. For a 'cognitively-plausible' model, on the other hand, requirements might include features of human learning like the ability to incrementally learn from small data or the ability to generalise to related but unseen tasks.

Model justification: no model is perfect and we advocate the investigation of diverse algorithmic solutions to a problem, even when they only partially fit a research problem. At the same time, the use of a given architecture should be justified with respect to the identified requirements. That is, in a spirit of transparency, it should be made explicit which desiderata are satisfied, and which are not.

Implementation: the model implementation should make clear *how* requirements are satisfied, and if relevant, at which level the algorithm simulates the given phenomenon: for instance, a lot of discussions have taken place around the cognitive implausibility of backpropagation in neural networks (Marblestone, Wayne, and Kording 2016). So a model can realistically implement a mechanism at a high-level while failing to reproduce the low-level. The present paper includes such an example, by implementing a mechanism inspired by neural adaptation in a statistical – not biological – fashion (§5.2).

Evaluation: it should finally be made clear *which* requirements will be experimentally evaluated. This includes raw performance on the task at hand, but also implementation-specific aspects such as interpretability of results or efficiency, as identified previously. Keeping the goal of artificial diversity in mind, not all requirements have to be evaluated or satisfied. The aim should be to understand which aspects of a given architecture give positive or negative results, with respect to a certain requirement.

3. Lexical acquisition and the Fruit Fly Algorithm

This paper investigates the **acquisition of lexical representations** in the usage-based framework of Distributional Semantics (DS: (Turney and Pantel 2010; Erk 2012)). In DS, the meaning of words is represented by points in a multidimensional space, derived from word co-occurrence statistics. Beyond the simplest, 'count-based' models of DS, a variety of more powerful approaches have been developed (Bengio et al. 2003; Pennington, Socher, and Manning 2014; Mikolov et al. 2013). State-of-the-art models perform remarkably well and are often a core component of NLP applications. Recent work on DS (e.g., ELMo: (Peters et al. 2018) and BERT: (Devlin et al. 2019)) shifts the scope of representations from word meaning to sentence meaning, pushing performance on specific, utterance-based tasks even further. The most successful work in the area is however oriented towards resource-rich engineering, and the present paper is instead concerned

with modelling human language. From this point of view, it becomes apparent that while the DS framework should be well-suited to implement usage-based approaches to semantic acquisition, it in fact fails at being a cognitively appropriate model, in the ways which we describe below.

As a starting point, we follow the **desiderata** highlighted in (Qasemizadeh, Kallmeyer, and Herbelot 2017) for a model of lexical learning: (A) high performance on fundamental semantic tasks, (B) efficiency, (C) low dimensionality for compact storage, (D) amenability to incremental learning, (E) interpretability. These considerations were not specifically brought up with cognitive plausibility in mind — it is for instance not so clear whether low dimensionality is a feature of cognition: see (Gorban, Makarov, and Tyukin 2020) for a discussion of the 'blessing of dimensionality'. Nevertheless, as we will see below, they point at crucial aspects of human language learning: the incremental process (D) that leads to the acquisition of full lexical competence (A), the robustness and efficiency of that process in spite of the poverty of the stimulus (B), and (to some extent) the ability of the speaker to formulate their linguistic knowledge in terms of explicit rules acting over categories (E).

With respect to these desiderata, the latest DS techniques can be seen to have multiple shortcomings. First, they require massive amounts of text, followed by computationally intensive procedures involving weighting, dimensionality reduction, complex attention mechanisms etc. The high complexity of most current architectures often comes at the cost of flexibility: once a language model is constructed, any new data requires a re-run of the complete system in order to be incorporated. This makes incrementality unsatisfiable in those frameworks (Sahlgren 2005; Baroni, Lenci, and Onnis 2007). Further, architectures themselves have become increasingly complex, at the expense of transparency. We recall that even Word2Vec (W2V: (Mikolov et al. 2013)), which is a comparatively simple system by today's standards, has attracted a large amount of literature which attempts to explain the effects of various hyperparameters in the model (Levy and Goldberg 2014; Levy, Goldberg, and Dagan 2015; Gittens, Achlioptas, and Mahoney 2017). Finally, high-performance DS representations are hardly or not at all interpretable. As a result, much research has been dedicated to producing representations that are intuitively interpretable by humans (Murphy, Talukdar, and Mitchell 2012; Luo et al. 2015; Fyshe et al. 2015; Shin, Madotto, and Fung 2018). These approaches typically attempt to preserve or reconstruct word labels for the dimensions of the dimensionality-reduced representations, but they can themselves require intensive procedures.

In the present paper, we will focus on requirements (A), (D) and (E). We note that (A) is a basic feature of human language and (D), more broadly, a fundamental attribute of animal cognition. Focusing on (D), it seems that we should find inspiration in algorithms from cognitive science, which in turn would allow us to derive interpretability (E) from the clear underpinnings of biological or psychological theories.

With this in mind, we propose to investigate the **Fruit Fly Algorithm (FFA)**. The FFA can be related to two existing techniques in computer science: Random Indexing and Locality-Sensitive Hashing . Random Indexing (RI) is a simple and efficient method for dimensionality reduction (Sahlgren 2005), originally used to solve clustering problems (Kaski 1998). It is also a less-travelled technique in distributional semantics (Kanerva, Kristoferson, and Holst 2000; Qasemizadeh, Kallmeyer, and Herbelot 2017; QasemiZadeh and Kallmeyer 2016). Its advocates argue that it fulfils a number of requirements of an ideal vector space construction method, in particular incrementality. As for Locality-Sensitive Hashing (LSH) (Slaney and Casey 2008), it is a way to produce hashes that preserve a notion of distance between points in a space.

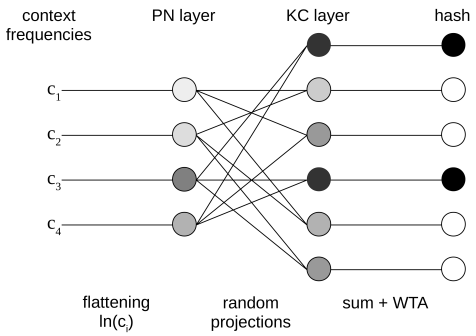

Figure 1
Schematic of the adapted FFA: (l=log, m=4, n=6, c=2, h=33.3). Darker cells correspond to higher activation; the dense representation of hashes has size 2.

In terms of **implementation**, the original FFA follows closely the biological architecture of the fruit fly. Our aim is to **evaluate** to what extent the algorithm can learn good lexical representations (A) while naturally implementing incrementality (D), and whether it satisfies some notion of interpretability (E).

4. Data

In the spirit of 'training small', the corpus used for our experiments is a subset of 100M words from the ukWaC corpus (Ferraresi et al. 2008), minimally pre-processed (tokenized and stripped of punctuation signs); this results in 87.8M words. Following common practice, we quantitatively evaluate the FFA as a lexical acquisition algorithm by testing it over the MEN similarity dataset (Bruni, Tran, and Baroni 2014), which consists of 3000 word pairs (751 unique English words), human-annotated for semantic relatedness. For our experiments, we compute two co-occurrence count spaces over our corpus, with different context sizes (± 2 and ± 5 around the target). We only consider the 10k most frequent words in the data, ensuring we cover all 751 words in MEN.

5. Model

The Fruitfly Algorithm mimics the olfactory system of the fruit fly, which assigns a pattern of binary activations to a particular smell (i.e., a combination of multiple chemicals), using sparse connections between just two neuronal layers. This mechanism allows the fly to 'conceptualise' its environment and to appropriately react to new smells by relating them to previous experiences. Our implementation of the FFA is an extension of the work of (Dasgupta, Stevens, and Navlakha 2017) which allows us to generate a semantic space by hashing each word – as represented by its co-occurrences in a corpus – to a pattern of binary activations. We first present the minimal adaptation along with the intuition behind the FFA and then introduce an extension in the form of a feedback mechanism that can be used in incremental settings.

5.1 The raw fruit fly

As in the original implementation, our FFA is a simple feedforward architecture consisting of two layers connected by random projections (Fig. 1).

The input layer, the *projection neuron layer* or *PN layer*, consists of m nodes $\{x_1 ... x_m\}$ corresponding to m context words. It encodes the raw co-occurrence counts of a target word with contexts, in a window of particular size. For instance, for a toy example where $m = 3$ and the three PNs correspond to contexts $\{meow, run, piano\}$, the word *cat* might be encoded as vector $[15, 5, 0]$, meaning that *cat* occurred 15 times in the context of *meow*, 5 times in the context of *run* and never in the context of *piano*. To satisfy incrementality over a expanding vocabulary, we additionally implement an expansion mechanism which enables m to be variable and grow as the algorithm encounters new data. Whenever an unknown context is observed, a node x_{m+1} is recruited to encode that context. Finally, in order to diminish unwanted effects resulting from the Zipfian distribution of natural languages (Zipf 1932), the first processing step converts 'raw' co-occurrence counts to their natural logarithm: $x_i = ln(c_i)$. This heuristic 'flattens' activation across the PN layer, reducing the impact of very frequent words (e.g., stopwords such as *the*, *or* and *of*).

The second layer (*Kenyon Cell layer* or *KC layer*) consists of n nodes $\{y_1 ... y_n\}$. It is larger than the PN layer and fixed at a constant size (n does not grow). PN and KC layer are *not* fully connected. Instead, each KC receives a constant number c of connections from the PN layer. This is initially achieved by performing sampling c times from a uniform distribution over the PN layer with $P(x_i) = \frac{1}{m}$ for $0 < i \leq m$ (without replacement). Most PNs will thus have about the same number of outgoing connections, but this number is variable. In other words, the mapping from PN to KC is a bipartite connection matrix $\mathbf{M}$ so that $\mathbf{M}_{ji} = 1$ if x_i is connected to y_j and 0 otherwise. The activation function on each KC is simply the sum of the activations of its connected PNs. In the final step, hashing is carried out via a winner-takes-all (WTA) procedure that 'remembers' the IDs of a small percentage $0 < h < 1$ of the most activated KCs as a compact representation of the word's meaning. So $WTA(y_i) = 1$ if y_i is among the hn highest values in y and 0 otherwise.

Note that, since both the connectivity per KC and the size of the KC layer are constant, the overall maximum number of connections is constant. Thus, the expansion mechanism (incrementing m) is designed to maintain that maximum. If the maximum is reached, the expansion mechanism samples existing PNs and reallocates outgoing connections of the sampled PNs to the new PN. The selection process is biased towards reallocating connections from those PNs with the most outgoing connections. This implements the tendency for even connectivity of the PN layer. This is important because two contexts with the same frequency, encoded as two PNs with the same level of activation, should have the same level of influence on the activations in the KC layer.

The expansion of dimensions from the PN layer to the KC layer in combination with random projections can be interpreted as a form of 'zooming' into a concept for a particular target word: multiple context words are randomly projected onto a single KC. If several of these context words are important for the target (i.e., their PNs have high activation), the corresponding KC will be activated in the final hash. We can imagine this process as aggregating dimensions of the original co-occurrence space, thus generating latent features which give different 'views' into the raw data. For example, one might imagine a random projection from the PNs *beak, bill, bank, wing*, and *feather* to a KC which is somewhat activated by the PNs *bank* and *bill* in finance contexts, and strongly activated for target words related to birds. Note that this behaviour lets us trace back

the most characteristic contexts for a particular target word, and gives interpretability to the KCs. We will come back to that feature in §8.

The FFA's hyperparameters are expressed as a 5-tuple (f, m, n, c, h), where f is the flattening function, m is the size of the PN layer (initially 0), n is the size of the KC layer, c is the number of connections leading to any one KC, and h is the fraction of activated KCs to be hashed.

5.2 Extension: an adaptation mechanism

Frequency effects have been shown to interfere with count-based models of lexical semantics. The "flattening" function of the minimal working FFA might not be strong enough to mitigate this issue. Traditional DS count-based models use weighting, often in the form of Pointwise Mutual Information (PMI), to decrease the importance of very frequent events in the generation of word vectors. Intuitively, PMI gives more weight to contexts which are *characteristic* for a target word: those that occur often with the target, but few times with other words. In neural implementations of DS like Word2Vec, a subsampling mechanism takes care of reducing the impact of frequent items. Both PMI and subsampling are effective techniques, but they are not suited to incremental systems. PMI is applied to a co-occurrence count matrix after an entire corpus has been read. Subsampling similarly relies on a preliminary analysis of the corpus, which returns a list of word probabilities. To respect our requirements, we must find a solution which is fully incremental.

We again find inspiration in the cognitive literature, and the mechanism of *neural adaptation*. This mechanism describes a decrease in response to a repeated or constant stimulus, and it has been subject to research for over a century, mainly in the visual domain, cf. (Stratton 1896; Webster 2012). (Wainwright 1999) proposes that neural adaptation serves the optimal transmission of information, allowing for a wider range of stimuli to be perceived (e.g., dark and bright scenes, silent and loud noises, subtle and strong odours). Neural adaptation is naturally incremental, occurring in time after a period or number of presentations of the stimulus. It is fair to assume that inconsequential events in a linguistic stimulus (e.g. very frequent events like the presence of determiners before nouns) should exhibit adaptation. We model such adaptation effects with a feedback mechanism which deletes connections between the PN layer and the KC layer based on the informativeness of each hash dimension. (For transparency, note that while this mechanism is cognitively plausible, our implementation is not and uses standard statistics.)

The feedback is applied in three steps. First, we analyse the set of hash sequences obtained from the FFA[1] and identify the set C of KCs which contribute the least to the discrimination of concepts. Second, we count how often each PN connects to a KC in C and how many overall connections it has. Third, we delete a connection between a PN and a KC in C if the PN connects to KCs in C more frequently than expected.[2]

1. Analysis of Hash Sequences. The intuition behind this feedback mechanism is to maximise discrimination between word vectors (see above comments on the role of neural adaptation for the discrimination of stimuli). We want FFA hashes to clearly

1 In analogy to the common practice of pilot task forces to de-brief after a flight, we propose to dub this feedback mechanism "de-briefing the Fruit Fly".

2 In the following we simplify: as PNs correspond one-to-one to context words, and so do KCs to hash dimensions, we use $\langle PN, context \rangle$ as well as $\langle KC, dimension \rangle$ interchangeably.

separate the meanings of two words along the relevant dimensions: for instance, cats and dogs share features as animals and pets, but they differ in behaviour, with dogs being on the whole more social creatures than cats. The patterns of latent concepts (KCs) representing two target words such as *cat* and *dog* depend on the KCs' connections to the PN layer (the contexts associated with the targets). So the particular way that PNs connect to KCs – the random projections – has an impact on the quality of the hashes. We hypothesise that ideally, the activation of a PN should only contribute to the activations of those KCs which actually help distinguish between concepts (and therefore assist their selection). Presumably, KCs which are selected in the WTA step either too rarely or too frequently do not provide full discriminative potential and are not informative enough; those correspond to latent topics which should be fine-tuned by the adaptation mechanism.

We quantify the informativeness of a hash position (henceforth: dimension) with the KL-divergence $D_{KL}(P|Q)$ of its observed average activation P to the average activation Q to be expected with a WTA procedure that selects h of KCs with uniform probability (h is one of the hyperparameters of the FFA; $0 < h < 1$).

A set of hash sequences would be maximally informative if each dimension y_j carried, on average, the same (maximum) amount of information. This is the case if $Q(y_j = 1) = h$ for any y_j.[3] Note that while Q is a uniform distribution, a good WTA procedure (i.e., one that minimises $D_{KL}(P|Q)$ for all dimensions) is *not* random uniform. Rather, the information that it receives through connections from the PN layer is restricted in a way that makes the procedure select certain KCs when they are useful and ignore them when they are not.

We confirm this reasoning with a sanity check, counting the 5K most frequent words of a 1M subset of the ukWaC corpus and applying an FFA = (f=log, m=4K, n=40K, c=6, h=0.05). It shows that while about 95% of the dimensions in a hashed space have $D_{KL}(P|Q) \leq 0.05$, some 2.5% of dimensions have $D_{KL}(P|Q) > 0.5$. In other words, some dimensions behave extremely differently from the optimum. Counting how often certain PNs connect to these highly divergent KCs, we find that the PNs with the highest number of such connections belong to stop words (*a, and, at, in, it, this* etc.). For example, over 80% of the connections going out from *and* lead to the 5% most divergent dimensions. The most plausible explanation is that stop words, with their extremely high frequency, overshadow the effects of other contexts and act as the sole contributors to the high activation of a KC, which is then constantly selected in the WTA step, making its dimension uninformative. The FFA will therefore benefit from "getting used" to these context words *if* they cause a certain KC to be selected too often.

Given a set of n-dimensional hash sequences, the analysis step first computes $D_{KL}(P_{y_j}|Q)$ for each dimension y_j, where P_{y_j} is the likelihood of y_j to be 0 or 1 and Q is its probability distribution of activation under the assumption that every y_j is maximally (and thus equally) informative. In practice, $Q(y_j = 1) = h$ for all y. These D_{KL} values make up the set V. We then select a set $C \subset V$ of the most divergent dimensions:

$$C := \{y_j \mid D_{KL}(P_{y_j}|Q) > \mu(V) + \sigma(V)\}. \quad (1)$$

3 This is subject to the assumption that dimensions are independent from each other, which is not true for all pairs of dimensions. This means that in practice, D_{KL} will never reach 0 for all dimensions, but it is still valid to serve as an objective.

If V was to follow the standard normal distribution, this would amount to approximately 15% of dimensions to be considered for adaptation; in reality, C will be smaller than this.

2. Selection of Candidates for Disconnection. Leaving the hash sequences and their dimensions behind and turning to the inside of the FFA, we interpret C as the set of KCs which are selected in the WTA step unusually often (or almost never). On the basis of C, the connection matrix **M** is used to obtain a frequency distribution $freq : X \mapsto \mathbb{N}$ which maps each PN $\{x_i, ..., x_m\}$ to the number of its connections to any of the KCs in C.

3. Disconnection. In the third step, the feedback mechanism decides which of the connections to KCs in C will be deleted. Of course, these highly diverging KCs usually also receive input from truly informative contexts; these connections should not be deleted. Instead, the preliminary experiment showed that uninformative PNs are characterised by a large number of connections to the KCs in C. Therefore we define a measure F_{over} which quantifies for each PN x_i its observed connectivity $c_o(x_i)$ to KCs in C relative to its expected connectivity $c_e(x_i)$. In other words, this measure of "overconnectivity" expresses how much more often a certain PN directly influences the WTA decision of an uninformative KC than expected.

The factor c_o is simply the relative number of connections to any KC in C: $c_o(x_i) = \frac{freq(x_i)}{|C|}$. As for c_e, we need to assume that the number of outgoing connections varies across the PN layer, because the feedback mechanism potentially deletes more connections than the expansion mechanism (cf. §5.1) will renew during the next round of co-occurrence counting. It is possible that at the time of the next round of feedback, the PN layer's connections are *not* evenly distributed. We therefore define a PN's expected connectivity to KCs in C as $c_e(x_i) = \frac{c(x_i)}{n}$, where $c(x_i)$ is that PN's number of outgoing connections. We thus calculate a PN's "overconnectivity" as

$$F_{over}(x_i) = \frac{c_o(x_i)}{c_e(x_i)} = \frac{freq(x_i)}{c(x_i)} \frac{n}{|C|}, \tag{2}$$

with C as defined in Eq. 1. The first term in Eq. 2 expresses the proportion of connections to uninformative KCs to KCs in general. The second term is constant per iteration of feedback and scales this proportion to the size of C. This allows to impose a threshold $t \in \mathbb{R}^+$ which decides for the deletion of any one PN's connections to the KCs in C if $F_{over}(x_i) \geq t$. A threshold of 1 is very conservative, imposing disconnections for any PNs with a c_o above average. The higher t, the more "lenient" the decisions to delete connections. We express this single additional hyperparameter t with a subscript (e.g., FFA_t).

6. Experiments

In order to characterise the behaviour and performance of our incremental FFA, we evaluate the quality of its output vectors against the MEN test set by means of the non-parametric Spearman rank correlation ρ. We first tune the hyperparameters of the minimal FFA over the counts (window size: ± 5) of the m=10K most frequent words of

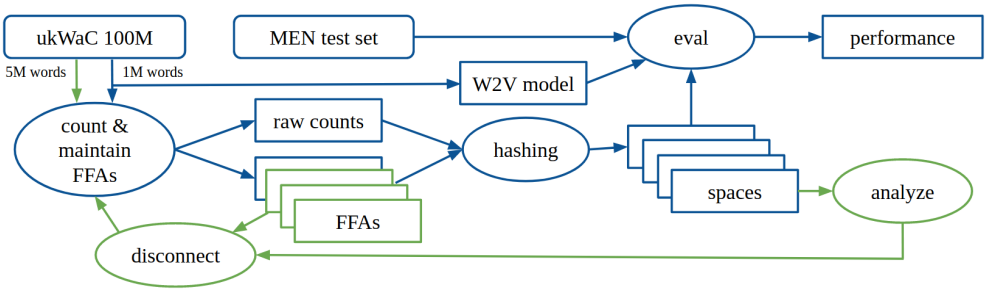

Figure 2
Processing steps of the experiments. Additional steps of the adaptation extension at the bottom in light green.

a held-out corpus. Our grid search returns the following optimal configuration: (*f*=*ln*, *m*=50, *n*=40000, *c*=20, *h*=0.08);[4] we use this for all further experiments. [5]

We first investigate the performance of the 'raw' FFA and then, in a second separate experiment on the same data, the behaviour of the FFA with adaptation mechanism (see Fig. 2 for a summary of the experimental design). For the first version, we *incrementally* generate a raw frequency-count model of the 10K most frequent words of our corpus, expanding the FFA with every newly encountered word. Every 1M processed words, the aggregated co-occurrences are hashed and the corresponding word vectors stored for evaluation. We compare a) the raw frequency space (input to the FFA); b) the final hashes (output of the FFA); c) a separate Word2Vec (W2V) model trained on exactly the same data, using standard hyperparameters and a minimum count set to match the 10K target words of our co-occurrence space. We repeat this experiment for window sizes ± 2 and ± 5.

The second experiment on the extended FFA is similar to the first one. We use the same hyperparameter configuration as before, but initialise multiple FFA_t with varying thresholds t for disconnection: $t \in \{0, 1, 2, 3, 4, 5, 7, 10, 15\}$, where FFA_0 does not carry out any disconnections. FFA_0 is expected to perform similarly to the FFA in the first experiment. Counting and expansion is carried out as previously, but only for a window size of ± 5. Furthermore, hashing is only carried out every 5M encountered tokens. The hash sequences are then analysed and feedback is applied directly before the next iteration of the incremental loop begins. We compare the various FFA_t to each other in terms of ρ-values, and the best FFA_t to the three modelling techniques in the first experiment (raw counts, minimal FFA, and Word2Vec).

7. Results

Fig. 3 shows the results of our first incremental simulation and, for comparison, the results of FFA_5 from the adaptation experiment. For the window size ± 5, we reach ρ = 0.245 for raw counts, ρ = 0.345 for the FFA output, ρ = 0.454 for FFA_5, and ρ = 0.600 for W2V. The 2-word-context setup yields very similar results (ρ*(FFA-2)* = 0.310; ρ*(counts-2)*

4 The grid search revealed in fact that the factor of expansion $\frac{n}{m}$ is minimally important. As this FFA is incremental, we start with m=50 and expand up to m=10K.
5 The source code for this implementation of the FFA, the extension, and the experiments is publicly available at `https://github.com/SimonPreissner/semantic-fruitfly`

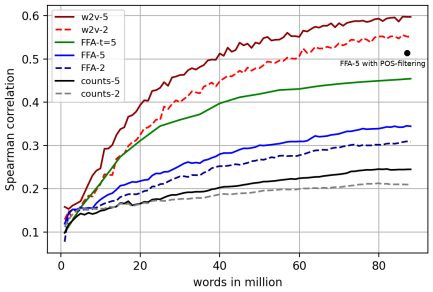

Figure 3
ρ-values of 'raw' counts, FFA-hashed spaces, FFA_5-hashed spaces, and W2V models (window sizes $\pm$ 2 (lines) and $\pm$ 5 (dotted)). The dot shows the performance with FFA-5 on POS-tagged data (nouns, verbs, and adjectives only).

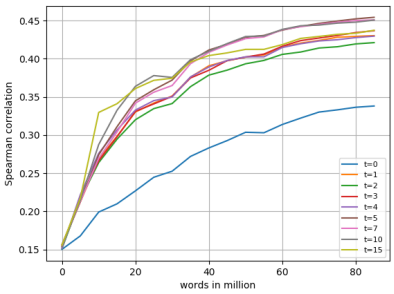

Figure 4
ρ-values of spaces by extended FFAs with varying thresholds t (FFA_0 does not apply feedback). Measures taken every 5M tokens of co-occurrence counting (window size $\pm$ 5).

= 0.210; *ρ(w2v-2)* = 0.555). The hashing by the minimal FFA thus has a clear and positive effect (+0.100 from 80M words on for the ± 5 setup). The amount of improvement is already visible after 20M of counted tokens (+0.05) and slowly increases with corpus size. Results are comparable to W2V for very small corpus sizes, but start lagging behind after 5M words. In comparison to the minimal FFA, the results for FFA_5 show much greater improvements over the raw counts (+0.205 from 80M words onward). Improvements are similarly large at the beginning but start lagging behind W2V later, after about 15M words.

Turning to comparisons among FFAs, Fig. 4 compares the ρ values for all FFA_t in the second experiment. There is a clear improvement in performance of all extended FFAs with respect to FFA_0, which (as expected) performs similarty to FFA-5 in Fig. 3. At the last iteration, the values range from ρ = 0.421 (+0.083, FFA_2) to ρ = 0.454 (+0.116, FFA_5). Among the extended FFAs, performances diverge quickly at the beginning and continue to improve at similar rates after about 30M words. In this early period, FFAs with higher thresholds tend to improve faster. The three best FFAs have thresholds of 5, 7, and 10; the most 'permissive' FFA_{15} falls behind with the others after the 30M token point. Note that there is a collective plateau of the learning curves at 35M words, which might stem from properties of the underlying text data.

Lastly, we investigate the dynamics of disconnection of the adaptation mechanism. Fig. 5 shows the number of affected PNs (i.e., PNs that lost at least one outgoing connection) per iteration of feedback. As assumed, the initial iterations affect an overwhelming number of PNs: at 5M words (first iteration) all FFAs apply disconnections to more than 66% of their PNs, except for FFA_{15}, which disconnects broadly in the second round. The number of affected PNs drops quickly to below 2000 (20%) from about 25M words on. At this point, two groups of FFAs emerge, similar to those in Fig. 4: FFAs with stricter feedback ($t \in \{1, 2, 3, 4\}$) fluctuate around 830 affected PNs with a slight downward trend, while FFAs with less adaptation ($t \in \{5, 7, 10, 15\}$) approach 0. For qualitative insights, Table 1 shows the overall number of deletions for some PNs of FFA_5. It confirms that the most affected PNs are indeed stopwords, but that there is also slight neural adaptation to content words.

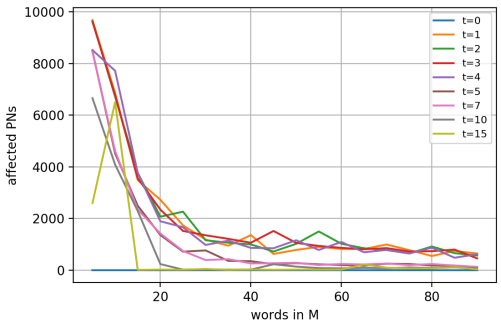

Figure 5
Number of PNs affected by the feedback mechanism

Table 1
FFA_5: overall number of deleted connections from a specific PN, with associated context; ranked by number of deletions.

Rank	#deleted	Context	PN
1	81	and	19
5	80	the	10
10	80	for	51
15	75	it	27
20	72	jobs.net	9993
100	27	back	255
200	16	got	121
500	12	marks	771
2000	9	1988	5714
5000	6	types	1422

8. Discussion

We now turn to a discussion of our results, focusing on the 'wish list' highlighted in §3.

Performance: hashing increases performance over the raw co-occurrence space by about 10 points overall. The minimal implementation is however outperformed by W2V after seeing around 5M words. When extended with the suggested feedback mechanism, performance gains over the baseline are twice as high, and this improvement especially takes place within the first 30M tokens encountered.

In the spirit of providing a comprehensive evaluation of the modelling power of the FFA, we attempt to pull apart aspects of the learning process that are captured by its very simple algorithm, and those that are not. In other words, which feature results in the clear increase over baseline performance? What does the original FFA fail to model with respect to W2V? Why does the adaptation mechanism improve the minimal working FFA and why do the various FFA_t behave slightly different from each other?

Starting with the original FFA, we know that the algorithm generates latent features out of the original space dimensions, encapsulated in each KC. We have tuned the size of the KC layer, so the number of features learned by the FFA should be optimal for our task. We assume that the performance displayed by the algorithm is due to correctly generalising over contexts. As for its *lack* of performance, we can make hypotheses based on what we know from other DS models. The minimal FFA does not perform any subsampling or weighting of its input data, and the log function we use to minimize the impact of very frequent items is probably too crude to fulfill that purpose. We can tackle this issue from the perspective of the data, by preemptively restricting the input. For example, when we informally inspect the performance of the algorithm on a POS-tagged version of our corpus, keeping only verbs, nouns and adjectives in the input and filtering some highly frequent stopwords (punctuation, auxiliaries), we obtain $\rho \approx 0.51$ over the whole corpus,[6] coming close to W2V's performance and thus indicating that indeed, a higher-level 'attention' mechanism could be added to the input layer. (Note

6 We use the top 4000 dimensions of the co-occurrence matrix (i.e., $m = 4000$), with $n = 16000$, $c = 20$ and $h = 0.08$.

that the olfactory system of actual fruit flies only has ≈ 50 odorant receptors, which makes it potentially less crucial to successfully suppress large parts of the input.)

Another approach to 'attention' is neural adaptation, whereby the response to an incoming signal decreases if there is continuous input and crucially, if that signal is not informative. This approach is modelled by our feedback mechanism: starting from an analysis of which components of a hash sequence help to discriminate it from others, we identify dimensions which are not informative. In order to render these dimensions (or within the FFA: KCs) more sensitive to incoming signals, we de-sensitivise them to those signals (i.e., activations of PNs) which persistently contribute to low informativeness (i.e., which are overly frequently connected to uninformative KCs). As in weighted neural networks, we achieve this by decreasing the weights associated to these connections. However, as we use a *bipartite* connection matrix **M**, 'decreasing the weight' of a connection amounts to deleting that connection altogether. The modelling of effects of neural adaptation in the feedback-extended FFA does not entirely exclude certain contexts from the hashing procedure; instead, it 'habituates' certain (but not all) KCs to them if the contexts are constantly uninformative. This allows 'habituated' KCs to focus more on other incoming signals and be a more informative part of hash sequences.

Concerning t, the threshold of disconnection, there are differences in performance gains of the habituated FFAs: a more 'lenient' FFA_t tends to learn faster in the beginning than its 'stricter' counterparts. On another note, the FFA_t with the highest threshold, FFA_{15}, initially learns considerably faster than the other FFAs, but falls behind other, 'stricter' FFAs after a while (cf. figure 4). Intuitively, a high threshold is beneficial in the beginning because there are only few contexts other than stopwords and very frequent content words can still positively influence the configuration of the FFA space when hashing certain concepts. At this stage, neural adaptation allows for better development if it is restricted to the most frequent contexts only. This changes once the counts of moderately frequent contexts reach a reasonable frequency and become the main driver for high-quality spaces. FFA_{15}, with the least strict feedback, probably misses this point in the course of learning, and continues to rely on the very frequent content words.

Incrementality: the FFA is fully incremental. Note that in our experiments, the W2V space is retrained from scratch after each addition of 1M words to the corpus while the FFA simply increments counts in its stored co-occurrence space. It is also in stark contrast with weighted count-based distributional models which require some global PMI (re-)computation to outperform the raw co-occurrence count vectors.

The adaptation mechanism is designed to preserve incrementality. In fact, while other models merely satisfy this characteristic, the extended FFA gives a strong incentive to process the available data incrementally. By pausing the construction of the count space (and the FFA alongside) at regular intervals and evaluating the reactions of the FFA to the data observed so far, the algorithm can adapt in order to better react to future developments of the count space. The incentive for incremental training is especially strong in the early phases in which the adaptation mechanism makes the most drastic changes to **M**. In our experiments, the advantage in learning speed of such extended FFAs over their minimal counterpart decreases a lot after about 30M words. Similarly to the development of human and animal cognition in which brain plasticity decreases after adolescence, it is plausible that the positive effect of the feedback mechanism can be maximised by altering the frequency of feedback over time.

The most prevalent effect of the adaptation mechanism in the extended FFA can be compared to the use of subsampling in Word2Vec: in both cases, the highly frequent context words are considered less important in the learning process so that other, more

informative contexts can have a greater influence. But Word2Vec achieves this in a non-incremental manner, by calculating a sampling probability for each word which depends on the word's overall corpus frequency. In contrast, the FFA's decision to put more or less attention on a context word is not made *a priori* on a statistical basis; instead, it is based on the observations of the dimensions in a hashed space.

One important benefit of the incremental hashes' analysis is that we have full control over the set of concepts that we feed to the adaptation mechanism. This means that in principle, we could decide to emphasise discriminability in a particular subspace of the embedding matrix in a dynamic fashion. For example, if the adaptation mechanism is provided a specific set of hash sequences belonging to the field of ornithology, certain context words like *wing* or *feather* which are normally considered informative may, in this domain, contribute relatively little to distinguishing between the given concepts. The adaptation mechanism will optimise the FFA for the small subspace that it is given, effectively carrying out neural adaptation to words like *wing* or *feather* which would otherwise not have been considered particularly *un*informative. With the ability to concentrate adaptation to a conceptual subspace, we are thus able to fine-tune the FFA to certain topics.

It is similarly useful to compare our FFA with Random Indexing (RI) which is, by nature, an incremental technique. We commented previously on the similarities between the two algorithms: the forward connections in the FFA can be seen as equivalent to the random vectors used in RI. However, while RI immediately learns vectors at reduced dimensionality, thereby combining co-occurrence counts and random projections, the FFA separates the quantitative aspect (i.e., the co-occurrences) from the qualitative aspect (i.e., the projections). This separation has a major advantage and a major disadvantage. The disadvantage is clear: it needs more memory to store the sparse co-occurrence matrix. The advantage, however, is that we can tune projections over the course of learning, motivating changes across time with an information gain objective. So again, the FFA comes out as a more dynamic solution.

Interpretability: the FFA's two-layer architecture is interpretable at every stage without any further computation. First, by following the forward connections from the PN layer to the KC layer, each KC (and therefore each hash dimension) can be expressed as the set of context words which influence the activity in the KC. Thus, by design of the FFA, each hash dimension has a finite set of labels which directly express the components of meaning associated with that hash dimension.

Second, given a hash signature, the FFA allows for uncomplicated back-tracing. Each of the activated nodes in a word's hash represents a single KC. The connections of these 'winner' KCs to the PN layer let us reconstruct which context words originally contributed to the largest activations in the KC layer. To illustrate this, we use the hashes obtained at the last iteration of our incremental experiments (based on window ± 5) and identify the $k = 50$ most characteristic PNs for each hash. In the case of the minimal FFA, we filter out stopwords and particles from these sets of characteristic PNs; for FFA_5, we report without filtering. Table 2 reports the characteristic PNs shared by various sets of input words. For the original FFA, for example, for the words *hawk, pigeon,* and *parrot* the *tailed, black, breasted, red,* and *dove* PNs are among the most influential, contributing to many of the activated KCs.

FFA_5 yields a similar, albeit differently ranked list of characteristic PNs for this particular set of word. Note however that for FFA_5 we do not need to filter stop words post-hoc, which is a convenient effect of the adaptation mechanism. With more 'attention' available to content words, some of the clusters obtained from the minimal FFA do

Table 2
Most characteristic PNs for selected sets of words. The importance of a PN for a word hash is estimated by the number of the PN's connections to KCs which are activated in the word's hash (window size $\pm$ 5).

Hashed Words	**Minimal FFA**	**FFA_5**
hawk, pigeon, parrot	tailed, breasted, black, red, dove	grey, crowned, tailed, red, seen, cuckoo
library collection museum	collection, national, new, art	libraries, national, board, royal, virtual data, collection, articles, description, main maritime, museum, science, war, articles
beard, wig	man, wearing, long, like, hair	n't, man, got, red, coat, off, hair, big, wearing
cold, dirty	get, said, war, mind	war, bad, cold, water, hands, enough, case

not form for FFA_5 because each of the words can be described with more characteristic contexts (e.g. the cluster *library, collection*, and *museum* in table 2). While reducing the impact of ubiquitous contexts, some 6% of characteristic PNs are still associated with tokens on the NLTK stopword lists, e.g. *n't* or *got*, both shared by *beard* and *wig*. This exemplifies the notion of *adaptation* whereby a constant stimulus often does not evoke a response except for those situations in which it becomes relevant.

In the same way that semantically related words can be grouped by this back-tracing of activations, we can connect *cold* to *dirty* in both the minimal and the extended FFA. Some of the shared important contexts of these two words seem to encode shared collocates (*cold/dirty war, cold/dirty mind, get cold/dirty*).

9. Conclusion

We started this paper suggesting that NLP should explore a broader variety of algorithms for its most fundamental tasks. We argued that such a diversity can give beneficial impulses to NLP, taking technical advances beyond 'raw' performance. We suggested that the natural world can serve as a source for inspiration. As illustration, we have explored what the olfactory system of a fruit fly can do for the representation of word meanings, by adapting it to the problem of incremental distributional semantics. Tested on a relatedness dataset, the original algorithm does capture latent lexical representation in the data above a simple co-occurrence baseline, and improves its performance greatly when modified with a mechanism inspired by neural adaptation. The overall performance score of the system lies below the state-of-the-art but in return, it provides natural incrementality and a high level of transparency.

In the spirit of porting the notion of biodiversity to 'artificial diversity', we highlight the elegance of the minimal random indexing process in the fruit fly, and its amenability to interpretation. We also hope that our system can pave the way for more efficient implementations with respect to computation and storage. We however also acknowledge that the minimalism of our FFA is insufficient to reach ideal performance in lexical acquisition. Future work should therefore focus on similar algorithms at a slightly higher level of complexity — or explore new and different approaches, bringing more diversity to NLP.

Acknowledgements

We thank the anonymous reviewers for their helpful comments on our original submission.

References

Baroni, Marco, Alessandro Lenci, and Luca Onnis. 2007. ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 49–56, Prague, Czech Republic, June. Association for Computational Linguistics.

Bender, Emily M. and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July. Association for Computational Linguistics.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bruni, Elia, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Dasgupta, Sanjoy, Charles F Stevens, and Saket Navlakha. 2017. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Erk, Katrin. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Ferraresi, Adriano, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54, Marrakech, Morocco, June.

Fyshe, Alona, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. 2015. A Compositional and Interpretable Semantic Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–41, Denver, Colorado, May. Association for Computational Linguistics.

Gittens, Alex, Dimitris Achlioptas, and Michael W. Mahoney. 2017. Skip-Gram - Zipf + Uniform = Vector Additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, Vancouver, Canada, July. Association for Computational Linguistics.

Gorban, Alexander N., Valery A. Makarov, and Ivan Y. Tyukin. 2020. High-dimensional brain in a high-dimensional world: Blessing of dimensionality. *Entropy*, 22(1):82.

Hovy, Dirk and Shannon L. Spruit. 2016. The Social Impact of Natural Language Processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, Berlin, Germany, August. Association for Computational Linguistics.

Kanerva, Pentii, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*, volume 22, Philadelphia, PA, August.

Kaski, Samuel. 1998. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 1, pages 413–418, Anchorage, AK, USA, May. IEEE.

Köhl, Maximilian A., Kevin Baum, Markus Langer, Daniel Oster, Timo Speith, and Dimitri Bohlender. 2019. Explainability as a Non-Functional Requirement. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 363–368, Jeju Island, South Korea, September. ISSN: 2332-6441.

Levy, Omer and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, Montreal, Quebec, Canada, December.

Levy, Omer, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Linzen, Tal. 2020. How Can We Accelerate Progress Towards Human-like Linguistic Generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online, July. Association for Computational Linguistics.

Luo, Hongyin, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2015. Online Learning of Interpretable Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1687–1692, Lisbon, Portugal, September. Association for Computational Linguistics.

Marblestone, Adam H., Greg Wayne, and Konrad P. Kording. 2016. Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10:94.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pages 3111–3119.

Murphy, Brian, Partha Talukdar, and Tom Mitchell. 2012. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1933–1950, Mumbai, India, December. The COLING 2012 Organizing Committee.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Preissner, Simon and Aurélie Herbelot. 2019. To be Fair: a Case for Cognitively-Inspired Models of Meaning. In *Proceedings of the Sixth Italian Conference on Computational Linguistics*, page 7, Bari, Italy, November.

QasemiZadeh, Behrang and Laura Kallmeyer. 2016. Random Positive-Only Projections: PPMI-Enabled Incremental Semantic Space Construction. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 189–198, Berlin, Germany, August. Association for Computational Linguistics.

Qasemizadeh, Behrang, Laura Kallmeyer, and Aurelie Herbelot. 2017. Projection Aléatoire Non-Négative pour le Calcul de Word Embedding / Non-Negative Randomized Word Embedding. In *Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 - Articles longs*, pages 109–122, Orléans, France, June. ATALA.

Sahlgren, M. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, Copenhagen, Denmark, August.

Shin, Jamin, Andrea Madotto, and Pascale Fung. 2018. Interpreting Word Embeddings with Eigenvector Analysis. In *32nd Conference on Neural Information Processing Systems (NIPS 2018), IRASL workshop*, Montréal, Canada, December.

Slaney, Malcolm and Michael Casey. 2008. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *IEEE Signal processing magazine*, 25(2):128–131.

Stratton, George M. 1896. Some preliminary experiments on vision without inversion of the retinal image. *Psychological Review*, 3(6):611.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July. Association for Computational Linguistics.

Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Wainwright, Martin J. 1999. Visual adaptation as optimal information transmission. *Vision Research*, 39(23):3960–3974, November.

Webster, Michael A. 2012. Evolving concepts of sensory adaptation. *F1000 Biology Reports*, 4, November.
Zipf, George K. 1932. *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press.