Blackbird's language matrices (BLMs): a new benchmark to investigate disentangled generalisation in neural networks

Anonymous ACL submission

Abstract

Current successes of machine learning architectures are based on computationally expensive algorithms and prohibitively large amounts of data. We need to develop tasks and data to train networks to reach more complex and more compositional skills. In this paper, we illustrate Blackbird's language matrices (BLMs), a novel grammatical dataset developed to test a linguistic variant of Raven's progressive matrices, an intelligence test usually based on visual stimuli. The dataset consists of roughly 48000 sentences, generatively constructed to support investigations of current models' linguistic mastery of grammatical agreement rules and their ability to generalise them. We present the logic of the dataset, the method to automatically construct data on a large scale and the architecture to learn them. Through error analysis and several experiments on variations of the dataset, we demonstrate that this language task and the data that instantiate it provide a new challenging testbed to understand generalisation and abstraction.

1 Introduction

All speakers can understand a sentence never heard before, or derive the meaning of a word or a sentence from its parts. And yet, these basic linguistic skills have proven very hard to reach by computational models. The current reported success of machine learning architectures is based on computationally expensive algorithms and prohibitively large amounts of data that are available for only a few, non-representative languages. To reach better, possibly human-like, abilities in neural networks' abstraction and generalisation, we need to develop tasks and data that help us understand their current generalisation abilities and help us train them to more complex and compositional skills.

Generalisation in NLP has been defined in a very narrow way, as extension from a set of data points to new data points of exactly the same nature (i.i.d. assumption). Not much effort has gone in trying to generalise to new problems or out of distribution (Schölkopf, 2019). Even under this very narrow definition, recent studies show that current algorithms do not generalise well (Belinkov and Bisk, 2018; Belinkov and Glass, 2019).

One likely reason why people generalise better is that they have a strong prior bias, grounded in the actual structure of the problem. A large body of literature of experimental work has demonstrated that the human mind is predisposed to extract regularities and generate rules from data, in a way that is distinct from the patterns of activation of neural networks (Lakretz et al., 2019a).

One possible approach to develop more robust methods, then, is to pay more attention to the decomposition of complex observations, discovering the factors in the generative process that gives rise to the data (Schölkopf et al., 2012). To study how to discover the underlying problem structure, machine learning research in vision has developed the notion of disentanglement. A disentangled representation can be defined as one where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors (Bengio et al., 2013).

To learn more disentangled linguistic representations, that reflect the underlying linguistic rules of grammar, we develop a new linguistic task. We will use a new set of progressive matrices tasks developed specifically for our goals and demonstrate their usefulness. While the use of automatically generated data in NLP is not new, this kind of progressive matrix generation for higher level linguistic reasoning has never been tried before for NLP, as far as we are aware.[1]

[1] The code and the data described in this paper will be released in full on publication. Currently, more examples of the data and more details on models specifications are to be found in the supplementary materials.

2 Blackbird's Language Matrices (BLMs)

Inspired by computational methods on vision, we develop a new linguistic task, to learn more disentangled linguistic representations that reflect the underlying linguistic rules of grammar.

The solution of the tasks requires identifying the underlying rules that generate compositional datasets (like Raven's progressive matrices), but for language. We call them Blackbird's Language Matrices (BLMs).

2.1 Progressive matrices for visual stimuli

Raven's progressive matrices (progressive because tasks get harder) are IQ tests consisting of a sequence of images (usually eight) connected in a logical sequence by underlying generative rules. The task is to determine the missing element (usually the last) in this visual sequence. An example and explanation of this task is given in Figure 1. The matrices are built according to generative rules that span the whole sequence of stimuli and the answers are constructed to be similar enough that the solution can be found only if the rules are identified correctly.

Traditionally, progressive matrices as intelligent tests are designed by hand, but recent research in vision that has used this task to train neural networks has typically employed some structured generative model to create larger numbers of questions (Wang and Su, 2015). In this way, a correct answer is consistent with the underlying generative model, so the learning process basically discovers how to induce the model. In this way, for example, it is possible to identify clear dimensions of successful and unsuccessful generalisation. For example, matrices for vision have shown that the best models can apply known abstract relationships in novel combinations, but fail in applying known abstract relationships to unfamiliar entities (Barrett et al., 2018).

2.2 Progressive matrices for language

Consider what subproblems need to be solved to reach the right answer in an RPM: (i) The elements manipulated by the rules must be identified. (ii) The relevant attributes of the manipulated elements must be identified. (iii) The rules of change of these attributes must be identified. (iv) The abstract structure of the matrix must be identified.

To instantiate these subproblems in the automatic generation of language matrices, first, we define the language tasks that need to be solved. Second, we define the rules governing the abstract automatic generation process and we compose rules into grammatical language templates. Finally, we automatically create large samples of data. We describe these steps below.

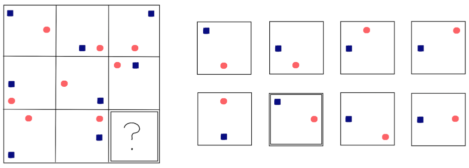


Figure 1: Examples of progressive matrices in the visual world. The task is to determine the missing element in a visual pattern: Given the matrix on the left, choose the last element of the matrix from the choice of elements on the right. The matrix is constructed according to two rules: Rule 1: from left to right, the red dot shape moves one place clockwise each time. This pattern continues onto the next row; Rule 2: from top to bottom, the blue square moves one place anticlockwise each time. This pattern continues onto the next column. Identifying these rules leads to the correct answer: the correct answer is marked by double edges; it is the only cell that continues the generative rules correctly.

2.2.1 Choosing the body of grammatical rules

We choose to construct data to determine if the rules of subject-verb number agreement can be learned and if other elements in the sentences that are involved (or not involved) in agreement can be identified. We choose to work on French because its agreement system, its verb conjugations and its noun phrase structure lends itself well to our investigation.

As a reminder, the main rule of subject-verb agreement in French, and English, states that subject and verbs agree in their number, and they do so independently of how many noun phrases intervene between the subject and the verb, as shown in the main clause examples of Figure 2. In practice, the intervening noun phrases can act as agreement attractors and trigger agreement mistakes, if they are close to the verb, like for example the fourth sentence in Figure 3.

Subject-verb agreement is a morphological phenomenon of appropriate complexity to start our investigations with BLMs. It is easily isolated from other aspects of a sentence, it is marked explicitly in the forms of words (for example by an $-s$ ending) and it does not depend on the words' meaning. Subject-verb agreement, then, is clearly limited to some specific words in the sentence, so that the

1	Subj-sing	N1-sing		Verb-sing
2	Subj-plur	N1-sing		Verb-plur
3	Subj-sing	N1-plur		Verb-sing
4	Subj-plur	N1-plur		Verb-plur
5	Subj-sing	N1-sing	N2-sing	Verb-sing
6	Subj-plur	N1-sing	N2-sing	Verb-plur
7	Subj-sing	N1-plur	N2-sing	Verb-sing
8	Subj-plur	N1-plur	N2-sing	Verb-plur

(a) Template

1	L' ordinateur	avec le programme		**est**	en panne.
2	Les ordinateurs	avec le programme		**sont**	en panne.
3	L' ordinateur	avec les programmes		**est**	en panne.
4	Les ordinateurs	avec les programmes		**sont**	en panne.
5	L' ordinateur	avec le programme	de l'expérience	**est**	en panne.
6	Les ordinateurs	avec le programme	de l'expérience	**sont**	en panne.
7	L' ordinateur	avec les programmes	de l'expérience	**est**	en panne.
8	Les ordinateurs	avec les programmes	de l'expérience	**sont**	en panne.

(b) Main clause

Figure 2: Agreement template and sentences generated by the templates. There are three types of context: main clause, completive clause, and relative clause. Template is exemplified with main clause. Translation of main: *The computer(s) with the program(s) of the experiment is/are not working*.

elements and the attributes manipulated by the underlying rules can be clearly identified. Moreover, agreement rules show structural properties, so that sequences of increasing complexity of application of the rule can be defined (Linzen et al., 2016; Linzen and Leonard, 2018).

2.2.2 Defining the BLM generative rules and creating the BLM templates

In describing the process to build BLMs, we will talk of *contexts*, the sequence of sentences whose last element needs to be identified, and *answer set*, the set of answers that instantiates the multiple choice task that needs to be solved by the BLM test.

Creating the templates contexts For the (semi)-automatic generation of the contexts, first, we define the abstract structure of the matrix as a progression in the number of attractors, as shown in Figure 2. The 'objects' that we manipulate are noun phrases and we manipulate their agreement attributes: the number of the head noun (which needs to match the number of the verb), the number of the closest noun and the number of the second noun (both can vary freely), as in Figure 2a.

So, for example, the sequence in Figure 2 is generated by a rule of progression of number of attractors (one and two), a rule of subject-verb agreement that alternates for every sentence between singular and plural of the head noun and a rule of number of the attractors that alternates between singular and plural every two sentences. Thus, the correct answer for this example is a sentence that has three attractors and a singular subject and singular attractors.

Thus, when learning rules of subject-verb agreement, the network needs to learn (i) to identify the relevant words that need to undergo agreement, the subject and the verb; (ii) needs to learn that attractors do not count for subject-verb agreement; (iii) needs nonetheless to pay attention to the attractors, and not simply ignore them, to be able to correctly predict which sentence will complete the matrix. To do so, as explained above, it will have to learn three separate rules that operate on the constituents of the sentence in different ways.

We generated sentences according to the rules with the adapted items from (Franck et al., 2002). To vary the structures, we also create three versions of the sentences: noun phrases in the matrix clause or embedded in a completive or a relative clause.[2] In total, there are 28 rules, and 896 matrices for each of the sentence types.

Creating the template answers The generation of possible answer sets is also a complex issue. Alternative (and incorrect) answers need to be sufficiently distinguishable, but also sufficiently similar to actually require the application of all three rules to be discarded. That is, the correct answer cannot be found by just learning some of the generative rules, instead of all of them, or by some simple heuristic. We opt for a choice among grammatical and ungrammatical alternatives comprising the choices exemplified in Figure 3, for the main clause structural context.

2.2.3 Creating the data sample

Once the BLM templates are defined, we need to create a large sample of natural, grammatical sentences, to train the networks and to generate the

[2] The completive corresponding to the example in Figure 2 is 'Je suppose que le(s) ordinateur(s) avec les programmes de l'expérience est/sont en panne', whose translation is *I assume that the computer(s) with the program(s) of the experiment is/are not working*. The corresponding relative clause is 'Le(s) ordinateur(s) avec le(s) programme(s) de l'expérience dont Jean se servait est/sont en panne', and its translation is *The computer(s) with the program(s) of the experiment that Jean was using is/are not working*.

Answers	Error type
1 L' ordinateur avec le programme et l'expérience est en panne.	Coord: the last noun appears in a coordination phrase
2 **L' ordinateur avec le programme de l'expérience est en panne.**	Correct: Correct answer
3 L' ordinateur avec le programme est en panne.	WNA: Wrong number of attractors
4 L' ordinateur avec le programme de l'expérience sont en panne.	AE: Agreement error
5 Les ordinateurs avec le programme de l'expérience sont en panne.	Alter N1: wrong number of the first NP
6 L' ordinateur avec les programme de l'expérience est en panne.	Alter N2: wrong number of the second NP

Figure 3: Different conditions of the answer matrix.

answer sets.

Creating natural contexts and answers We use an automatic method, based on context-aware word embeddings, specifically we use Camembert (Martin et al., 2020) to generate more sentences. The general process consists in masking some of the nouns to generate other most probable nouns and use them to construct new sentences. More specifically, we mask different nouns in the three kinds of constructions. For the main clause, we masked the first noun simultaneously for all the sentences in a matrix, and generated the five most probable nouns. We applied the same procedure for the second noun. So one template matrix can give rise to ten matrices. For the completive clause and relative clause, we mask the head noun in the subject and main verb, as well as first noun and second noun of the embedded clause, and applied the same procedure for the main clause. One template matrix can generate twenty more matrices. We generate lexical variants of answers by the same procedure used in the contexts.

Creating data variants: Matrices with lexical variation All the matrices described above were created with the same lexical items in each matrix. To force the learner to concentrate on the grammatical rules, sentences of the same type are shuffled and lexically varied matrices are created as shown in Figure 4, as an example of a main clause matrix.

Creating data variants: Unordered matrices Natural language tasks and problems often are not limited to a single sentence but span over several sentences, for example, in textual entailment, machine translation, reference resolution, dialogue. It is therefore important to build test sets that stress language learning abilities and from which we can extrapolate realistic conclusions about real learning patterns. To test whether the ordered nature of the data we build is important, we also shuffle each basic matrix, to keep the templates constant, but make the order random.

With these novel datasets, we train learners and study their ability to learn the underlying rules giving rise to subject-verb agreement. As illustrated by Figure 4, this task can get quite difficult. We explore whether current models can learn to perform the task at all, and what factors of variation help the learning of the underlying rules.

3 Learning the matrices

To demonstrate that we can learn these matrices, we train several models. The computational choices of the problem concern the representation of the data set, the representation of the actual sentences and sequence of sentences and, finally, the computational architecture. We describe these methodological components below.

Data and embeddings The training data consists of 42800 BLMs (sequences of 7 context sentences and the corresponding correct continuation), split into 90% for training and 10% for validation. For testing, we have 4688 sequences of 7 sentences as BLMs and 6 possible answers for each sequence. To obtain representations of our data, we use FLAUBERT, a transformer model for French (Le et al., 2020), pretrained using a masked language modeling objective similar to BERT (Devlin et al., 2019). As this model gives us representations in context for each word in a sentence, we create an average representation for our sentences, so that we have a single vector for the entire sentence.[3] Batch size is of 354 sentences.

3.1 Variational Information Bottleneck with disentanglement

We test a variant of β-variational autoencoders for disentanglement. Higgins et al. (2016) propose an approach to learning disentangled representations inspired by the human tendency to reduce redundancy. To learn statistically independent factors,

[3] In these averaged representations, we omit the special tokens (e.g. $[CLS]$, $[SEP]$) added by the transformer at the beginning and end of a sentence. In addition, we pad each sentence to have the same length as the longest sentence in the data. This padding consists in adding 0s to the representation of the sentence.

Contexts	
Example	Translation
1 La conférence sur l'histoire a commencé plus tard que prévu.	*The talk on history has started later than expected.*
2 Les responsables du droit vont démissionner.	*Those responsible for the right will resign.*
3 L' exposition avec les peintures a rencontré un grand succès.	*The show with the paintings has met with great success.*
4 Les menaces de les réformes inquiètent les médecins.	*The threats of reforms worry the doctors.*
5 Le trousseau avec la clé de la cellule repose sur l'étagère.	*The bunch of keys of the cell sits on the shelf.*
6 Les études sur l'effet de la drogue apparaîtront bientôt.	*The studies on the effect of the drug will appear soon.*
7 La menace des réformes dans l' école inquiète les médecins.	*The threat of reforms in the school worries the doctors.*
Answers	
Example	Translation
1 Les nappes sur les tables et le banquet brillent au soleil.	*The tablecloths on the table and the console shine in the sun.*
2 **Les copines des propriétaires de la villa dormaient sur la plage.**	*The friends of the owners of the villa were sleeping on the beach.*
3 Les avocats des assassins vont revenir.	*The laywers of the murderers will come back.*
4 Les avocats des assassins du village va revenir.	*The lawyers of the murderers of the village will come back.*
5 La visite aux palais de l' artisanat approchent.	*The visit of the palace of the crafts is approaching.*
6 Les ordinateurs avec le programme de l' expérience sont en panne.	*The computers with the program of the experiment are broken.*

Figure 4: Example of lexically varied contexts for the main clause contexts. Correct answer in bold.

that is a disentangled representation, a constraint is added to the loss function of a variational autoencoder that forces closeness to the prior and embodies pressure for redundancy reduction and latent factors independence (the D_{KL} factor in the equation 1 below). Higgins et al. (2017) propose the addition of a single hyperparameter β to the original framework to limit the capacity of the latent variable and control the rate of learning of independent factors. A β-VAE with $\beta = 1$ corresponds to the original VAE, while a β-VAE with $\beta > 1$ is pushed to learn more efficient latent representations. [4]

3.2 Our model

We apply this general framework to our language data, with a substantial modification that makes our model more similar to a variational information bottleneck (VIB) approach (Alemi et al., 2017) than a proper VAE, as described and motivated below. A picture of the architecture is shown in Figure 5.

Encoder and decoding classifier The encoder is composed of four one-dimensional CNNs, three fully connected layers and ReLU activation.[5] The structure of the decoding classifier is the mirror image of the encoder, but with a different output dimensionality, as shown in Figure 5. It is composed of three fully connected layers, four one-dimensional transposed CNNs and ReLU activation.[6]

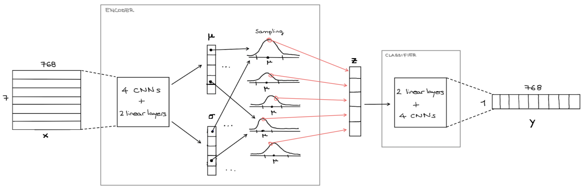


Figure 5: Model architecture

Loss function An adapted version of the objective function of the β-VAE objective is used. The β-VAE loss is composed of two different terms: the reconstruction loss and the Kullback-Leibler divergence. The reconstruction loss is the binary cross-entropy between the output of the model and the true answer. It penalizes the output based on the distance from the true answer. The Kullback-Leibler divergence is used to measure the divergence of two probability distributions. In our model, it puts pressure to reduce redundancy and reduce the distance from a prior.

In our approach, instead of reconstructing the original input (the sequence of sentences), the model produces a new and compressed representation of the input. The intuition is that the latent

[4] However, Higgins et al. (2017) warn us that a high value of β can result in a trade-off between reconstruction and disentanglement. In fact, having $\beta > 1$ leads to poorer reconstructions when compared to the original VAE. They show that their β-VAE model learns disentangled representation whose generating factors are as good as ground truth generating vectors, but the quality of reconstruction is not as good as entangled representation (images are less crisp).

[5] Structure and dimensionality: Input: 1x768x7 ; Encoder: four CNNs 1x100x7 (stride 1), three fully-connected layers 1x300; Output: 1x10 (dimensionality of the latent vector).

[6] Structure and dimensionality: Input: 1x10 (latent vector); Encoder: three Fully connected layers of dim 1x300, four Transposed-1d-CNN of dim 1x100x1 (stride 1); Output: 1x768x1 (new representation of the input).

layer, having learned the important factors of the input, can be used as is to produce a new single vector which represents the sequence and is close to the answer. The new vector will then be compared, at training time, with the correct answer. The model is updated after an entire sequence of seven sentences and the loss function is computed between the produced vector and the true answer vector.

So, in our model, x and y are not the same, as y is not the reconstruction of x. Instead, the output y is a vector of real numbers. In binary cross entropy —the loss function we use— these numbers are interpreted as a distribution over $0, 1$.

Equation 1 shows our objective function, where x is our input, z is the latent variable, β is our disentangled hyperparameter and D_{KL}, the Kullback-Leibler divergence, is the regulariser that forces the solution to be close to the prior, and $p(z)$ is a Gaussian prior, q_ϕ refers to the encoder and p_θ refers to the decoder, and L_{BCE} is binary cross-entropy loss.

$$(1) \quad \mathcal{L}(\theta, \phi; x, z, \beta) = \mathbb{E}_{q_\phi(z|x)}[L_{BCE}(f_\theta(z), y)] - \beta D_{KL}(q_\phi(z|x) \| p(z))$$

4 Experimental results

We perform several experiments to investigate how the properties of the dataset support the learning of subject-verb agreement and what latent representations are developed. First we investigate if the sentence type (main, completive or relative clause) has an effect on the results and errors. In terms of correct answers, main clauses are actually the hardest ones to learn. For all the three datasets, we observe roughly the same distributions in terms of correct answers and types of errors. In the following analysis of results and errors, then we no longer distinguish by sentence type.

Experimental results by β and z values Figure 6 presents results for the three data sets: the basic data set, the lexically-varied data set and the reshuffled data set. For high values of β and large sizes of the latent vector there is hardly any learning, so we will not discuss them further.

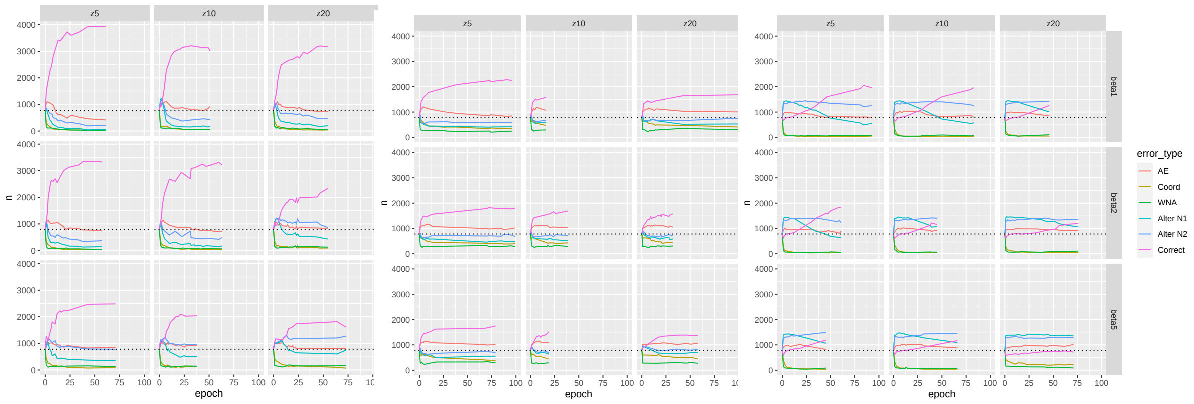
For the more successful configurations, we can observe, first of all, that the learning curve is steep, showing already a good rate of learning after a few epochs, if the latent vector is small, reaching a good 84.2% accuracy in the best, and easiest, case. $\beta = 1$ yields the best results, with a certain consistency of results across latent spaces of different sizes, the smallest the best. $\beta = 1$ is the value of a normal VIB that does not force disentanglement. This result then confirms what already found in the vision literature that entangled representations lead to better accuracy (Higgins et al., 2017).

If we class the data to train the models as ± lexical uniformity and ± sequence ordering, we can see that the basic data is the easiest model (+ lexical uniformity and + sequence ordering); the lexically varied data is harder (− lexical uniformity and + sequence ordering); the shuffled data is the hardest(− lexical uniformity and − sequence ordering). This indicates that sequence ordering does provide information the models are able to use, while lexical variation does not help identifying the underlying formal invariants of the examples. This latter result replicates findings in vision that the best models can apply known abstract relationships in novel combinations, but fail in applying known abstract relationships to unfamiliar entities (Barrett et al., 2018).

Error analysis Figure 6 also shows interesting patterns of errors, which differ between the two models with sequence order (basic and lexically varied) and the model without (shuffled). The different errors indicate ability of the models to learn different types of information: subject-verb agreement requires long-distance, structural information; errors on N1 and N2 tell us whether the model exhibits recency effects, thereby showing, like humans, that both structural and linear considerations come into play in learning agreement; choosing the wrong number of attractors is a very salient form of structural deviance from the correct answer and coordination is a more subtle one.

For sequence order models, agreement errors are always the most frequent, followed by N1 and N2 alternatives, while coordination and number of attractors mistakes occur much less frequently, suggesting the models do learn the difference in construction and the rule of attractor sequence. This result matches our intuitions that these are also the two most saliently different cases from the right answer, because they differ in structure.

For the model without sequence order, the results are overall worst, indicating that providing the sequence actually helps in finding the right answer. The pattern of errors, though, is a little different, with most errors on N1 and N2, thereby showing



(a)
Template matrices

(b)
Lexical variation matrices

(c)
Shuffled matrices

Figure 6: Results on test set for the three models. Dotted horizontal random baseline.

Rule	Expected error
(R1) Subj-verb agreement: The subject and the main verb agree. They alternate between singular and plural at each sentence.	Increase in agreement errors (AE).
(R2) Number of attractors: The number of attractors grows with increments of one.	Increase in coordination and number of attractors errors (Coord and WNA).
(R3) N1 grammatical number: The number of the first attractor alternates between singular and plural in increments of two.	Wrong number of the first attractor NP (alter N1).
(R4) N2 grammatical number: The number of the second attractor never changes.	Wrong number of the second attractor NP (alter N2).

Table 1: Predictions of expected errors for masking of latent factors.

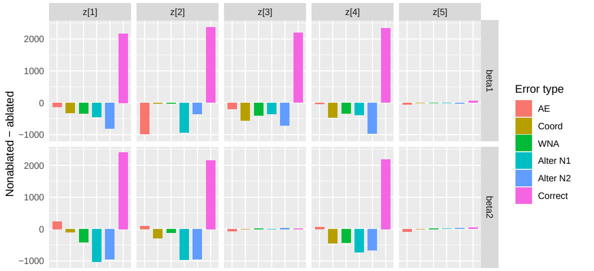


Figure 7: Error analysis by ablation of latent components. The y-axis shows the difference in errors between the non-ablated and the ablated model for each ablated component of the latent vector z, when $z = 5$.

linear considerations dominate learning if ordering is not provided; coordination and number of attractors mistakes are still much rarer, suggesting the model does learn salient differences in structure.

Analysis of disentanglement by ablation of latent factors Recall that disentangled representations are those where the single latent units are sensitive to changes in single generative factors. It can be determined whether the latent representations are disentangled by ablation studies that modify portions of the latent vectors. These should give rise to predictable effects. BLMs contexts are built by generative rules and the answers set was chosen according to specific patterns. So assuming we have a latent space that encodes the underlying generative rules, we can make the predictions shown in Table 1.

The process to test these expectations in error patterns consists in learning the latent vector in the usual way, then masking one of the latent vector components (putting its value to 0) and running the classifier with this partially-masked vector. Analysis of errors on the test set by ablated latent component are shown in Figure 7.

We can observe that all ablated models do overall worse than the non-ablated one, and also that component $z[5]$ is not useful, for $\beta = 1$, which already shows that the information is factorised. As expected, for $\beta > 1$, more factors are not useful, $z[3]$ and $z[5]$. For the errors, no clear pattern emerges. If we look at what kind of error is majoritarily degraded, we can say that agreement errors are markedly degraded if $z[2]$ is masked in $\beta = 1$, and $z[4]$ appears to be associated to the rule on number of attractors (R2).

Discussion The experiments show that BLMs define a hard, but learnable task. Their structured construction lends itself to controlled experiments, and the task is challenging enough that the models make informative patterns of mistakes. Our models can learn BLMs, and show that both lexical uniformity and sequence information help learn the right solution to the tests, but that the best results are still at least partially entangled. Finding models which learn disentangled representations of this task is a challenging open problem, which can help our understanding of different deep learning architectures.

5 Related work

The current paper does not have any direct comparison, as, to our knowledge, this is the first proposal of a dataset for language using BLMs. But it is inspired by and situated among work on disentanglement and generalisation for vision, where RPMs datasets have been used, and it contributes to the investigation on learning of agreement by neural networks.

Disentanglement datasets for vision and language van Steenkiste et al. (2020) develop a dataset for vision to learn tasks similar to Raven's Progressive Matrices, and evaluate the usefulness of the representations learned for abstract reasoning tasks. They observe that disentangled representations enable quicker learning using fewer samples. RPMs and their language equivalent have not been used before for language, as far as we know, but one other dataset exists to learn disentanglement for language, dSentences (M'Charrak, 2018).[7] Like our dataset, it is large in terms of size, but, unlike our dataset, the examples are unrealistically simple. The factorial combinations of very simple sentences with a few morphological marking does not constitute a sufficiently realistic challenge from the linguistic point of view. Moreover, natural language tasks and problems often span over several sentences, as discussed above, so the ordered sequence that characterises our BLMs is a crucial difference.

Related work on disentanglement for vision and language In the literature on disentanglement for vision, Higgins et al. (2016) and related work propose an approach to variational autoencoders based on redundancy reductions, and pressure to learn statistically independent factors. The disentangled representations enable zero-shot learning and emergence of visual concepts (Higgins et al., 2018). Following work shows the conditions when representations that align with underlying generative factors of variation of data emerge in optimisation (ELBO bound) (Burgess et al., 2018), and demonstrates that inductive biases are necessary to learn, but can be successfully encoded in potentially imprecise and incomplete labels (Locatello et al., 2020). Mercatali and Freitas (2021) extend VAEs to learn discrete representations appropriate for language. The proposed model outperforms continuous and discrete baselines on several qualitative and quantitative disentanglement benchmarks and extrinsic evaluations.

Related work on learning agreement Previous work on agreement has tested recursive neural network (RNN) language models and found that RNNs can learn to predict English subject-verb agreement, if provided with explicit supervision (Linzen et al., 2016). Follow-up work has shown that RNNs are better at long-distance agreement if they can use large vocabularies to form rich lexical representations to learn structural patterns Bernardy and Lappin (2017). Gulordava et al. (2018) extends previous work to four languages of different linguistic properties (Italian, English, Hebrew, Russian) and shows the models make accurate predictions and compare well with humans, thereby suggesting that the networks learn deeper grammatical competence. Recent work by Lakretz et al. (2019b) studies RNNs in more detail, looking at single neurons, and finds that individual neurons encode linguistically meaningful features very saliently and propagate subject-verb number agreement information over time.

6 Conclusions

In this paper, we have introduced Blackbird's language matrices (BLMs), a novel linguistic dataset, generatively constructed to support investigations in representation learning of grammatical rules. Through error analysis and several experiments on variations of the dataset, we demonstrate that this language task and the data that instantiate it provide a new testbed to understand generalisation and abstraction.

The contribution of the paper lies in the definition of a new challenging task, the development of its actual data and of a general procedure to develop many other such datasets, on different linguistic problems. But it also lies in tackling a mixture of language tasks and reasoning to take us closer to investigations of human linguistic intelligence.

[7] This dataset is similar to the dSprites dataset for vision (https://github.com/deepmind/dsprites-dataset))

References

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *Proceedings of the International Conference on Learning Representations (ICLR) 2017*.

David G.T. Barrett, Felix Hill, Adam Santoro, Ari S. Morcos, and Timothy Lillicrap. 2018. Measuring abstract reasoning in neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations (ICLR)*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transaction of the ACL*, 7:49–72.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798– 1828.

Jean-Philippe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*, 15(2):1–15.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-vae. In *2017 NIPS Workshop on Learning Disentangled Representations*.

Jacob Devlin, 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. Association for Computational Linguistics.

Julie Franck, Gabriella Vigliocco, and Janet. Nicol. 2002. Subject-verb agreement errors in French and English: The role of syntactic hierarchy. *Language and Cognitive Processes*, 17(4):371–404.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. 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 1195–1205. Association for Computational Linguistics.

Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. 2016. Early visual concept learning with unsupervised deep learning.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. 2018. Scan: Learning hierarchical compositional visual concepts.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019a. The emergence of number and syntax units in LSTM language models. *arXiv preprint arXiv:1903.07435*.

Yair Lakretz, Germán Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019b. The emergence of number and syntax units in LSTM language models. *CoRR*, abs/1903.07435.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Tal Linzen and Brian Leonard. 2018. Distinct patterns of syntactic agreement errors in recurrent networks and humans. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. 2020. Disentangling factors of variation using few labels. In *Eighth International Conference on Learning Representations - ICLR 2020*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Amine M'Charrak. 2018. Deep learning for natural language processing (NLP) using variational autoencoders (VAE).

Giangiacomo Mercatali and André Freitas. 2021. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of EMNLP 2021*.

Bernhard Schölkopf. 2019. Causality for machine learning. Technical report, arXiv:1911.10500v2.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. 2012. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK.

Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. 2020. Are disentangled representations helpful for abstract visual reasoning? In *NeurIPS 2019*.

Ke Wang and Zhendong Su. 2015. Automatic generation of Raven's progressive matrices. In *Procs of IJCAI 2015*.