BLM-AgrF: A New French Benchmark to Investigate Generalization of Agreement in Neural Networks

Aixiu An[1]*, **Chunyang Jiang**[2], **Maria A. Rodriguez**[3]*, **Vivi Nastase**[2] and **Paola Merlo**[2]

[1]Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology, aixiu@mit.edu

[2]Department of Linguistics
University of Geneva
{Chunyang.Jiang, Paola.Merlo}@unige.ch, vivi.a.nastase@gmail.com

[3] Computer Science and Information Technology
Lucerne University of Applied Sciences and Arts, maria.anduezarodriguez@hslu.ch

Abstract

Successful machine learning systems currently rely on massive amounts of data, which are very effective in hiding some of the shallowness of the learned models. To help train models with more complex and compositional skills, we need challenging data, on which a system is successful only if it detects structure and regularities, that will allow it to generalize. In this paper, we describe a French dataset (BLM-AgrF) for learning the underlying rules of subject-verb agreement in sentences, developed in the BLM framework, a new task inspired by visual IQ tests known as Raven's Progressive Matrices. In this task, an instance consists of sequences of sentences with specific attributes. To predict the correct answer as the next element of the sequence, a model must correctly detect the generative model used to produce the dataset. We provide details and share a dataset built following this methodology. Two exploratory baselines based on commonly used architectures show that despite the simplicity of the phenomenon, it is a complex problem for deep learning systems.

1 Introduction

Over the last years, driven by the surge in deep learning methods, models in NLP have become very powerful. They have even reached superhuman performance on standard benchmarks such as SuperGLUE (Wang et al., 2019) and SQuAD (Rajpurkar et al., 2018). Deeper probing, though, shows that this is due to the models' surprisingly robust superficial natural language understanding ability. This indicates that new benchmarks are necessary that accurately show the level of progress in language processing (Ruder, 2021).

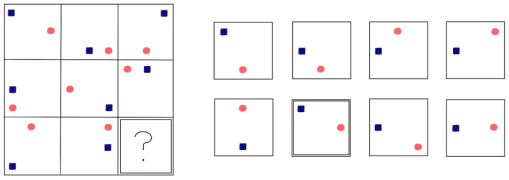


Figure 1: An example Raven's progressive matrix (best seen in colour). The matrix is constructed according to two rules: (i) the red dot moves one place clockwise when traversing the matrix left to right; (ii) the blue square moves one place anticlockwise when traversing the matrix top to bottom. The task consists in finding the tile in the answer set that correctly completes the sequence, indicated with a double border.

In this paper, we describe such a benchmark, developed based on a new method described in (Merlo et al., 2022), and summarized in Section 2. The method defines a procedure for building datasets that capture specific linguistic phenomena in a structured problem, inspired by the visual patterns detection tasks in Raven's progressive matrices (RPM) (Raven, 1938), as in Figure 1.

The visual RPMs manipulate elements with attributes such as position, shape, colour and size. The language matrices manipulate phrases, dependencies in the syntactic tree, and lexical, grammatical and semantic attributes between connected elements of a sentence. To successfully tackle such a complex problem, a system must detect structure and patterns in the data. To complement the structure of the problem, the candidate answers set is

*The work was done while the author was at the University of Geneva.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1363–1374
May 2-6, 2023

also specifically designed – the negative answers are built following specific criteria – to help determine which facets of the problem the system is able to learn, and which it is not.

By enabling different levels of analysis, from the solution of the task in different controlled setting to the analysis of the errors, this dataset intends to support the development of neural models with stronger abilities of abstraction and generalization, and more complex and compositional skills, that could learn robust models from few examples, and ultimately be deployed on low-resource languages. The code and the data are available here: https://github.com/CLCL-Geneva/BLM-SNFDisentangling.

2 BLM-AgrF: Blackbird's Language Matrices for agreement

	CONTEXTS TEMPLATE			
1	NP-sing	PP1-sing		VP-sing
2	NP-plur	PP1-sing		VP-plur
3	NP-sing	PP1-plur		VP-sing
4	NP-plur	PP1-plur		VP-plur
5	NP-sing	PP1-sing	PP2-sing	VP-sing
6	NP-plur	PP1-sing	PP2-sing	VP-plur
7	NP-sing	PP1-plur	PP2-sing	VP-sing
8	NP-plur	PP1-plur	PP2-sing	VP-plur

	ANSWER SET	
1	NP-sing PP1-sing et NP2 VP-sing	Coord
2	**NP-plur PP1-plur NP2-sing VP-plur**	correct
3	NP-sing PP-sing VP-sing	WNA
4	NP-sing PP1-sing PP2-sing VP-plur	AE
5	NP-plur PP1-sing PP1-sing VP-plur	WN1
6	NP-plur PP1-plur PP2-plur VP-plur	WN2

Figure 2: BLM instances for verb-subject agreement, with two attractors. WNA= wrong number of attractors; AE= agreement error; WN1= wrong nr. for 1^{st} attractor noun (N1); WN2= wrong nr. for 2^{nd} attractor noun (N2).

The data format we present has been called Blackbird's Language Matrices (BLMs) (Merlo et al., 2022), because it requires the presentation of the linguistic phenomenon of interest in the form of a complex set of sentences that have both syntagmatic and paradigmatic relations, thereby, like in the RPM visual version, forming a matrix structure.

A BLM has a structure defined by a combination of rules. The starting point is defining the linguistic problem that needs to be learned (e.g. subject-verb agreement) and the grammatical rules that define it. The combination of rules can be complex and each rule can act as an interfering factor obfuscating the other rules. The next step is to devise the rules governing the abstract automatic generation process that embody the properties of the linguistic phenomenon and its underlying rules. Combining these examples of grammatical rules will produce templates that can be used to automatically create large samples of data with lexical/structural variety. To allow for probing the learned model, apart from the correct answer, the answer sets contain negative examples built by corrupting some of the generating rules. This helps investigate the kind of information and structure learned, and the type of mistakes a system is prone to.

2.1 BLM-AgrF for subject-verb agreement

The BLM-AgrF dataset we illustrate here defines implicitly the rules of subject-verb agreement in French. As a reminder, the main rule of subject-verb agreement in French, and English, states that subjects and verbs agree in their number. Agreement is a rule that applies to the structure of the sentence and not the linear order, so agreement applies independently of how many noun phrases intervene between the subject and the verb.

Subject-verb agreement is a morphological phenomenon of appropriate complexity to start our investigations with BLMs. Subject-verb agreement is clearly limited to some specific words in the sentence so that the elements and the attributes manipulated by the underlying rules can be clearly identified. It is marked explicitly in the forms of words (for example by an $-s$ ending) and it does not depend on the words' meaning. 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). We choose to work specifically on French because its agreement system, its verb conjugations and its noun phrase structure lend themselves well to our investigation.

The data that illustrates this linguistic rule must show all patterns of combination of agreement between subject and verb but also include data that illustrate the structural nature of the rule. Noun phrases inserted in the subject NP as prepositional complements or relative clauses act as intervening elements. We consider one and two such noun phrases, to increase the distance between the subject and the verb, and different clause complexities to produce data that covers syntactic structures of various depths.

While intervening noun phrases do not enter into

an agreement relation based on the grammatical rules, in practice, the intervening noun phrases can act as agreement attractors and trigger agreement mistakes, if they are close to the verb. More specifically, Franck et al. (2002) show, in experiments with French and English speakers, that attraction is determined by the syntactic distance between an intervening noun and the head noun.

BLM-AgrF grammatical templates To generate the BLMs for the subject-verb number agreement, we develop a context-free grammar based on the targeted linguistic phenomenon and the interfering factors chosen, illustrated in Figure 3.

```
<CONSTRUCTION>→ <AGREEMENT>

# structure
<AGREEMENT>→ <MAIN-CLAUSE>
<AGREEMENT>→ <COMPLETIVE-CLAUSE>
<AGREEMENT>→ <RELATIVE-CLAUSE>

# L' ordinateur avec le programme est en panne .
<MAIN-CLAUSE>→
        <SUBJNP(Num)><ATTRACTORS><VP(Num)>

# L' ordinateur avec le programme dont Jean se servait est
en panne .
<RELATIVE-CLAUSE>→
        <SUBJNP(Num)><ATTRACTORS>
            <RELCLAUSE><VP(Num)>

# Jean suppose que l' ordinateur avec le programme est
en panne .
# Jean suppose que l' ordinateur avec le programme de l'
expérience est en panne .
<COMPLETIVE-CLAUSE>→
        <COMPCLAUSE><MAIN-CLAUSE>

# e.g.: "dont Jean se servait"
<RELCLAUSE>→ {Rel chunks}
# e.g.: "Jean suppose que"
<COMPCLAUSE>→ {Comp chunks}

# e.g. ["L' ordinateur", "Les ordinateurs"]
<SUBJNP(Num)>→ {NP chunks (Num)}

# e.g. PP1: ["avec le programme", "avec les programmes"]
# e.g. PP2: ["de l' expérience", "des expériences"]
<ATTRACTORS>→ <PP>
<ATTRACTORS>→ <PP><ATTRACTORS>
<PP>→ {Prep chunks (attractor)}

# e.g.: ["est en panne", "sont en panne"],
<VP(Num)>→ {VP chunks (Num)}
```

Figure 3: Context-free grammar for the subject-verb agreement in French, illustrated with examples.

The agreement between the subject and the verb is explicitly included in the production rule for the different types of clauses. The different types of clauses will lead to sentences with different structures, and the attractors' rule will insert one or two NPs between the subject and the verb to create different levels of linear and syntactic distance between them.

To instantiate these templates, our starting point are the examples in Franck et al. (2002, appendix 1). They provide a set of subject NPs of various complexity – including prepositional phrases, themselves of various complexity. We produced sentences based on these subject NPs by manually adding verb phrases, and by making the NPs more complex to increase the distance between the subject and the verb in the sentence. Each of these sentences is used to produce a *seed*. A seed contains all number variations of the NPs and VP in the sentence, and additional complement or prefix phrases to produce sentence variations of increased complexity. The end result is a set of 32 seeds, which provide the terminals (chunks) for the grammar. This progression from Franck et al. (2002) examples to seeds is illustrated in Figure 4.

The rules for BLM-AgrF generation The grammatical templates generate individual sentences. It is shown in Figure 2. The rules for a BLM-AgrF problem generation use the grammatical templates to generate a sequence of sentences according to specific sentence attributes. In our case, the sentence attributes that vary are the grammatical number of the subject and verb ({sg, pl}), the number of attractors ({1, 2}), and the grammatical number of each of these attractors ({sg, pl}). For each of the three clause types in the grammar, varying the above-mentioned attributes process will generate eight sentences, as illustrated in Figure 5.

To avoid biases in the process that may be caused by having an overly consistent input (i.e. the sequence following always the same sentence structure), different sentence sequences are generated for the same seed and same clause type by varying the alternation of values for the chosen attributes. For the sequences shown in Figure 5, the subject-verb grammatical number alternates between {sg, pl}, sentences with one attractor are generated first with alternating grammatical number, and then sentences with two attractors are built, but using a fixed grammatical number for the second attractor. We can alternate instead between {pl, sg}, generate sentences with two attractors first, and use a different grammatical number for the second attractor. Considering all these variations, we can obtain $\binom{2}{2} \times \binom{2}{2} \times \binom{3}{2} = 24$ different sequences for each

generation seed and each clause type (illustrated in more detail in Appendix A.1, Figure 10).

Each of these sequences of eight sentences will be transformed into a problem. The first seven sentences will be the input (we call it *context*), and the eighth will be included in the answer set.

The BLM-AgrF answer set The answer set will contain the eighth sentence as the correct answer, and another five candidates generated by corrupting one of the generating rules. The candidates in the answer set are generated such that they are distinguishable from the correct answer but relevant and challenging. Unlike other RPM-like datasets in vision, we choose to have a fixed set of answer types, to be able to then do a type-based error analysis. At the bottom of Figure 5, we present the answer set for the *Main clause* sequence in the table, with the one correct and five incorrect answers, and the characteristics of the wrong answer candidates.

The BLM-AgrF dataset The dataset consists of lexical instantiations of the grammatical templates produced based on the linguistic phenomenon – a subject-verb agreement in French declarative sentences – in simple and complex structures, thanks to noun phrases of various lengths and complexity between the subject and the verb in the sentence.

The manually provided seeds are useful to control the structure of the sentence and to ensure a starting point with syntactically and semantically valid sentences. By applying the rules of BLM-AgrF generation to the 32 seeds, which generate 24 sequences for each seed and for each clause type, we obtain a first dataset consisting of 2304 BLM-AgrF problems. We call this dataset *type I*.

To introduce some lexical variation in this dataset in a semi-automatic manner, we use CamemBERT (Martin et al., 2020) to replace individual words in the sentences in the type I dataset. We mask different words in the three types of clauses one at a time, and generate the five highest probability replacements that will be substituted in the sentence sequence and the candidate set:

Main clause : mask the subject noun and second noun in the sentence, e.g.:

Les MASK avec le programme de l'experience sont en panne.

Les ordinateurs avec le MASK de l'experience sont en panne.

Completive clause : mask the subject and verb in the completive clause, and mask the nouns in the embedded clause, e.g.:

MASK suppose que les ordinateurs avec le programme de l'experience sont en panne.

Jean MASK que les ordinateurs avec le programme de l'experience sont en panne.

...

Relative clause mask the head noun and verb in the relative clause, and the subject noun and following noun in the main clause, e.g. :

Les ordinateurs avec le programme de l'experience dont MASK se servait sont en panne.

Les ordinateurs avec le programme de l'experience dont Jean se MASK sont en panne.

...

The process is illustrated in Figure 6, and a more detailed view of masking scenarios is shown in Figure in Appendix A.2.

By applying these lexical variations on the type I dataset we obtain a dataset containing 38400 BLM-AgrF problems. We call this dataset *type II*. To further increase the lexical variation, we build the *type III* dataset, where a BLM-AgrF problem consists of a combination of sentences (with the same grammatical structure) from different type II problems. As this dataset consists of resampled sentences from type II, it will also contain 38400 BLM problems.

These three datasets are split 90:10 into train and test subsets. During experiments we take a random 0.1 portion of the training set for validation.

3 Experiments

Transformers and other neural architectures have shown very high performance on a variety of NLP tasks. We describe here two baselines to investigate the difficulty in learning the underlying regularities of subject-verb agreement on the proposed dataset. Figure 7 shows the general process flow. Each sentence in the input is encoded separately using a pre-trained multilingual transformer model,[1] which were shown to capture, among others, syntactic information (Hewitt and Manning, 2019).

[1]The results reported here are based on sentence embeddings obtained using BERTTokenizer and BERTModel from the *transformers* Python library, using the pretrained *BERT-base-multilingual-cased* model `https://huggingface.co/bert-base-multilingual-cased`. This encoder produces an embedding of size 768 for each sentence. We have run preliminary experiments with French-specific sentence embeddings using FlauBERT (Le et al., 2020). The results were lower than when using a multilingual cased BERT language model.

Example subject NPs from Franck et al. (2002)
L'ordinateur avec le programme de l'experience The computer with the program of the experiments
Manually expanded and completed sentences
L'ordinateur avec le programme de l'experience est en panne. The computer with the program of the experiments is down.
Jean suppose que l'ordinateur avec le programme de l'experience est en panne. Jean thinks that the computer with the program of the experiments is down.
L'ordinateur avec le programme dont Jean se servait est en panne. The computer with the program that John was using is down.

A seed for language matrix generation					
Jean suppose que Jean thinks that	*l'ordinateur* the computer	*avec le programme* with the program	*de l'experience* of the experiment	*dont Jean se servait* that John was using	*est en panne* is down
	les ordinateurs the computers	*avec les programmes* with the programs			*sont en panne* are down

Figure 4: Examples from Franck et al. (2002), manually completed and expanded sentences based on these examples, and seeds made based on these sentences for the subject-verb agreement BLM-AgrF dataset that contain all number variations for the nouns and the verb.

Main clause						
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.
Completive clause						
1	Jean suppose que	l'ordinateur	avec le programme			est en panne.
2	Jean suppose que	les ordinateurs	avec le programme			sont en panne.
3	Jean suppose que	l'ordinateur	avec les programmes			est en panne.
4	Jean suppose que	les ordinateurs	avec les programmes			sont en panne.
5	Jean suppose que	l'ordinateur	avec le programme	de l'expérience		est en panne.
6	Jean suppose que	les ordinateurs	avec le programme	de l'expérience		sont en panne.
7	Jean suppose que	l'ordinateur	avec les programmes	de l'expérience		est en panne.
8	Jean suppose que	les ordinateurs	avec les programmes	de l'expérience		sont en panne.
Relative clause						
1		L'ordinateur	avec le programme		dont Jean se servait	est en panne.
2		Les ordinateurs	avec le programme		dont Jean se servait	sont en panne.
3		L'ordinateur	avec les programmes		dont Jean se servait	est en panne.
4		Les ordinateurs	avec les programmes		dont Jean se servait	sont en panne.
5		L'ordinateur	avec le programme	de l'expérience	dont Jean se servait	est en panne.
6		Les ordinateurs	avec le programme	de l'expérience	dont Jean se servait	sont en panne.
7		L'ordinateur	avec les programmes	de l'expérience	dont Jean se servait	est en panne.
8		Les ordinateurs	avec les programmes	de l'expérience	dont Jean se servait	sont en panne.

Answer set for problem constructed from lines 1-7 of the main clause sequence		
1	L'ordinateur avec le programme et l'experiénce est en panne.	N2 coord N3
2	**Les ordinateurs avec les programmes de l'experiénce sont en panne.**	correct
3	L'ordinateur avec le programme est en panne.	wrong number of attractors
4	L'ordinateur avec les programmes de l'experiénce sont en panne.	agreement error
5	Les ordinateurs avec le programme de l'experiénce sont en panne.	wrong nr. for 1^{st} attractor noun (N1)
6	Les ordinateurs avec les programmes des experiénces sont en panne.	wrong nr. for 2^{nd} attractor noun (N2)

Figure 5: BLM-AgrF instances for verb-subject agreement, with two attractors (programme, experiénce), and three clause structures. And candidate answer set for a problem constructed from lines 1-7 of the main clause sequence.

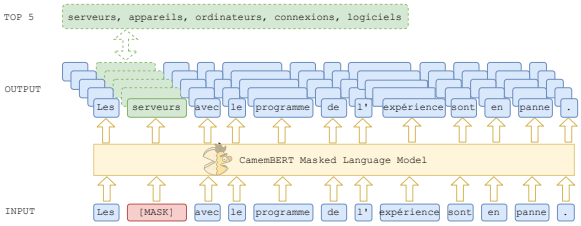


Figure 6: Creation of lexical variants by generating variations of a masked input using CamemBERT (Martin et al., 2020)

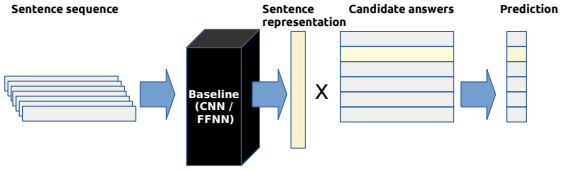


Figure 7: Illustration of the baseline setup experiments.

3.1 Models

We use two baseline systems – a feed-forward neural network (FFNN) and a convolutional neural network (CNN). Because the sentence embedding produced by the transformer captures structural information and we are presenting sentences in a sequence, both the FFNN and the CNN will have the chance to find patterns shared across the sentences.

The input to the FFNN is a concatenation of the sentence embeddings in the sequence (size 7 * 768), that is passed through 3 fully connected layers that gradually compress the input (7 * 768 $\xrightarrow{layer1}$ 3.5 * 768 $\xrightarrow{layer2}$ 3.5 * 768 $\xrightarrow{layer3}$ 768) to the size of a sentence representation. Because of the full connectedness between successive layers, the FFNN has the capacity of capturing patterns spread out over the entire input vector.

The input to the CNN is an array of embeddings, of size (7 x 768). This is passed through three successive layers of 2-dimensional convolutions, with a kernel size (3x3) (stride 1, no dilation). The output of the convolution is passed through a fully connected layer to compress it to the sentence representation size (768). Because of the kernel size, stride=1, and no dilation, this setup will focus on finding localized patterns in the sentence sequence array. If the NPs and verb grammatical numbers are encoded in a more localised manner within the sentence representation, this architecture should detect the patterns in the sequence.

The output of the two networks is the same – a vector representing the sentence embedding of the correct answer. The learning objective is to maximize the probability of the correct answer from the candidate answer set. Because the incorrect answers in the answer set are specifically designed to be minimally different from the correct answer, we implement the objective through the max-margin loss function. This function combines the distances between the predicted answer and the correct and erroneous ones. We first compute a score for the embedding e_i of each candidate answer a_i in the answer set $\mathcal{A}$ with respect to the predicted sentence embedding e_{pred} as the cosine of the angle between the respective vectors:

$$score(e_i, e_{pred}) = cos(e_i, e_{pred})$$

The loss uses the max-margin between the score for the correct answer e_c and for each of the incorrect answers e_i:

$$\mathcal{L}_a = \sum_{e_i} [1 - score(e_c, e_{pred}) + score(e_i, e_{pred})]^+$$

At prediction time, we take the answer with the highest $score$ value from a candidate set as the correct answer.

3.2 Results and discussion

The results of the experiments, in terms of F1 averages over 5 runs, are shown in Figure 8, and the detailed version is in Appendix A.3. The experiments were run on a VM on the Google Cloud Platform with one NVIDIA Tesla T4 GPU and 8G memory. We ran experiments for 50 epochs, with a learning rate of 0.001 and Adam optimizer. On type II and type III data, a run took 20 minutes, on type I data 2 minutes.

As a reminder, Type I data is lexically consistent – the same vocabulary is used in all sentences in the sequence, and in the answer candidates. Type II has a limited amount of lexical variation – one word in each sentence is different. Type III is more lexically varied, with little, if any, lexical overlap between any of the context or answer candidate sentences.

The models perform well when using the full amount of training data – the heatmaps in the left column in Figure 8 –, confirming that the experimental setup used is suitable for benchmarking the problem. Type I data has available 2073 instances for training (of which 20% are used for validation).

Figure 8: F1 averages over 5 runs, for the FFNN and CNN baselines, when training on all the available training data, or on the same amount of data (2073 instances)

When training models on this amount of data for type II and type III data, the performance on these subsets drops dramatically – the heatmaps on the right in Figure 8. This indicates that the success on Type I data is due to finding superficial clues, which do not generalize to data with higher lexical variability in both the input and the candidate answers. Overall, the FFNN model performs better, potentially indicating that the interesting patterns are not localized, but rather more spread out in the inputs. With different stride and dilation parameters, the CNN might improve its performance.

A plot of the different error types made by the different models (relative to the size of the test data) is presented in Figure 9. The error plots show how the frequency of different types of errors changes when using a model trained on data of a different type than the test data.

The error analysis presented in Figure 9 reveals several insights about the data and the performance of the models.

The different errors indicate the 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.

Across both models, the highest error is the N2_alt – the wrong number in the second attractor, the one closest to the verb. This proximity preference suggests that the models are rather shallow, with linear distance exerting greater influence rather than syntactic distance. In one case, when tested on type III data, the two models diverge. The pattern of results might indicate that CNNs find more localised patterns that allow them to avoid a recency bias. 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 intuition that these are also the two most saliently different cases from the right answer because they differ in structure.

These results and error analysis show that curated datasets like the one presented here reveal the superficiality of the positive results on the main task. If the underlying structural rules of the subject-verb agreement had been learned properly, lexical variation would not prove so disrupting and recency effects would not be as strong.

4 Related work

The current paper does not have any direct comparison, as this is the first proposal of a dataset for language using a BLM scheme. But it is inspired by work on generating RPM problems, and on solving such problems in computer vision, and it contributes to the investigation on learning of agreement by neural networks.

Structured datasets for vision and language The automatic generation of RPM-like matrices, whether in vision or in language, is challenging, technically, in two aspects. First, how do we represent the RPM problems to tackle their variations, regularities, and irregularities? Second, how do we ensure that the generated RPMs are valid?

To overcome these challenges, some efforts have been made in computer vision: Wang and Su (2015) formulate RPMs with first-order logic, which have inspired Barrett et al. (2018) who propose Procedurally Generated Matrices (PGM) dataset through relation-object-attribute triple instantiations. Zhang et al. (2019) use the Attributed Stochastic Image Grammar (A-SIG, proposed by Zhu and Mumford, 2006) as the representation of RPM and create the Relational and Analogical Visual rEasoNing (RAVEN) dataset.

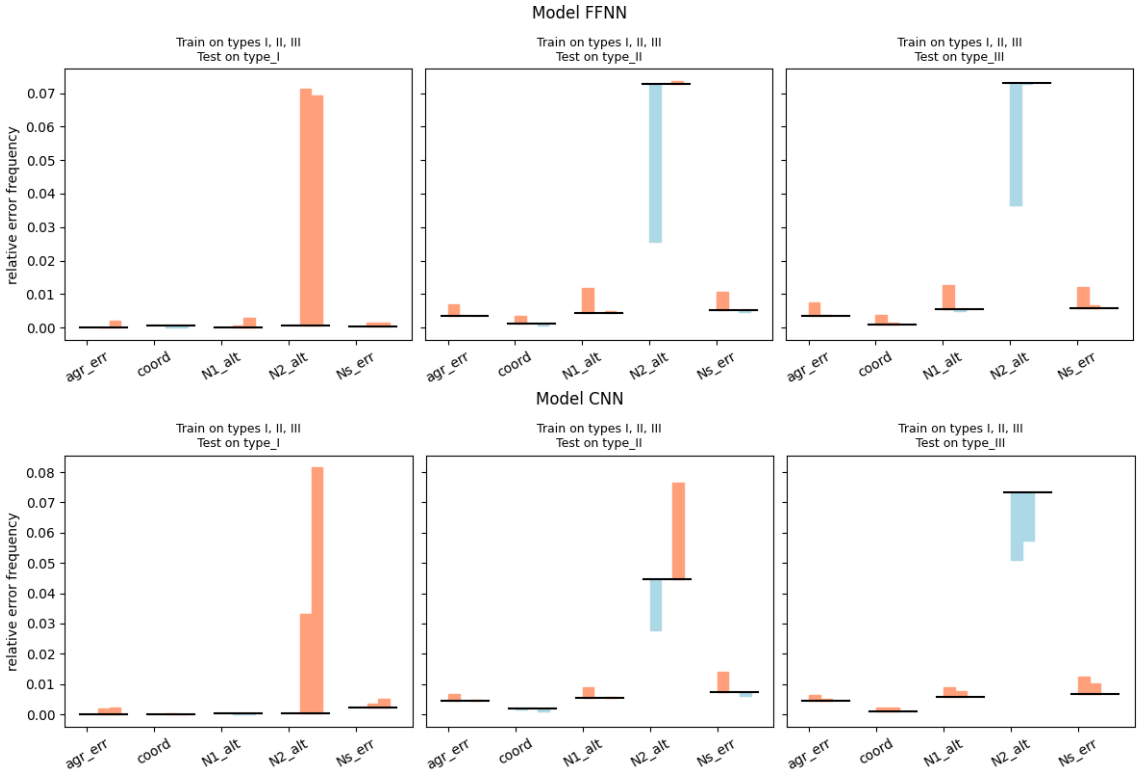


Figure 9: Relative frequencies of error types (relative to test data size) made by the different models, using models trained for all types of data. The reference – training and testing on the same data type – are given as the black lines, and we plot as vertical bars the increase (orange) or decrease (blue) in error when using a model trained on the other two data types.

These structured datasets have been mostly developed to study issues of generalisation and disentaglement. van Steenkiste et al. (2020) developed a dataset for computer vision similar to the RPMs, and evaluate the usefulness of the representations learned for abstract reasoning. They note that learning disentangled representations leads to faster few-shot learning. M'Charrak (2018) developed a large dataset, consisting of simple examples containing a few morphological markings. They use this dataset to learn disentangled sentence representations. The simplicity of the sentences does not provide a sufficiently realistic challenge from a linguistic point of view.

Learning agreement Previous work on agreement has tested recurrent 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). Bernardy and Lappin (2017)'s follow-up work has shown that RNNs are better at modeling long-distance agreement if they can train the model on top of a corpus where a larger (10000 types vs. 100) vocabulary is used – the rest of the words are replaced by their POS to highlight structural patterns. Gulordava et al. (2018) explore the RNNs capacity to track abstract hierarchical structure, by predicting long-distance number agreement in various constructions in four languages (English, Hebrew, Italian, Russian). Their results suggest that RNNs can learn hierarchical grammatical phenomena and not just shallow patterns. Lakretz et al. (2021) found that individual neurons in an RNN can encode linguistically meaningful features, and propagate subject-verb number agreement information over time. In a recent paper, Li et al. (2023) investigates deeper representational issues, by contrasting two kinds of agreement, subject-verb agreement and past-participle agreement in French. They argue, based on theoretical accounts, that these superficially similar kinds of agreement, involve in fact very different abstract operations and demonstrate that transformers do reflect this difference in their representations.

5 Conclusions

In this paper we have introduced BLM-AgrF, an instance of Blackbird's Language Matrices (BLM)

(Merlo et al., 2022). This novel linguistic dataset is generatively constructed to support investigations in representation learning of grammatical rules. Each instance, consisting of a sequence of sentences and a candidate answer set, was built using a combination of rules, to provide a layered and structured dataset for learning more complex models. The various layers of the dataset allow for a variety of explorations, from disentangled sentence representations to capture structure and regularities within a sentence, to modular architectures that could capture structure and regularities in the sentence sequences. The purposefully built candidate answers supports more in-depth analyses of the behaviour of a system, and provide insights into the source of prediction errors.

Experiments using baseline set-ups – feed-forward networks and CNNs – show that the task is difficult for previously successful sentence representations and neural architectures, despite the fact that the agreement rule they are supposed to discover is rather simple. This supports our hypothesis that the task the data embodies could provide a new benchmark for modeling generalization and abstraction.

6 Limitations

Manual creation for seeds for the synthetic data The seeds to generate the data are manually chosen, and the grammar rules are specifically designed for the problem. The process may be further automated in the future through higher-level formalisation of the matrix generation process.

Language variations and linguistic phenomena The dataset described in this paper focuses on subject-verb agreement in French, with the main verb in the present tense, covering common interfering factors, different clause complexities and various depth of syntactic structures. While the simplicity of the modeled rule can be perceived as a limitation, it was a deliberate feature. The low performance of the transferred models to different test sets shows that the simple rule was not easy to model. But as systems become successful on a given dataset, new, more complex versions can be built with richer phenomena at various linguistic levels, including in morphologically-rich languages.

Human upper bounds We do not have, in this paper, an explicit experiment on human upper bounds for the different data types.

Acknowledgments

We gratefully acknowledge the partial support of this work by the Swiss National Science Foundation, through grants #51NF40_180888 (NCCR Evolving Language) and SNF Advanced grant TMAG-1_209426 to PM.

References

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. 2018. Measuring abstract reasoning in neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4477–4486. PMLR.

Jean-Phillipe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. In *Linguistic Issues in Language Technology, Volume 15, 2017*. CSLI Publications.

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, New Orleans, Louisiana. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*.

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.

Bingzhi Li, Guillaume Wisniewski, and Benoît Crabbé. 2023. Assessing the Capacity of Transformer to Abstract Syntactic Representations: A Contrastive Analysis Based on Long-distance Agreement. *Transactions of the Association for Computational Linguistics*, 11:18–33.

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*.

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). Master's thesis, ETH Switzerland.

Paola Merlo, Aixiu An, and Maria A. Rodriguez. 2022. Blackbird's language matrices (BLMs): a new benchmark to investigate disentangled generalisation in neural networks. *ArXiv*, cs.CL 2205.10866.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

John C. Raven. 1938. Standardization of progressive matrices. *British Journal of Medical Psychology*, 19:137–150.

Sebastian Ruder. 2021. Challenges and Opportunities in NLP Benchmarking. `http://ruder.io/nlp-benchmarking`.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ke Wang and Zhendong Su. 2015. Automatic generation of raven's progressive matrices. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 903–909. AAAI Press.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. 2019. RAVEN: A dataset for relational and analogical visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5317–5327. Computer Vision Foundation / IEEE.

Song-Chun Zhu and David Mumford. 2006. A stochastic grammar of images. *Found. Trends Comput. Graph. Vis.*, 2(4):259–362.

A Appendix

A.1 Generation process

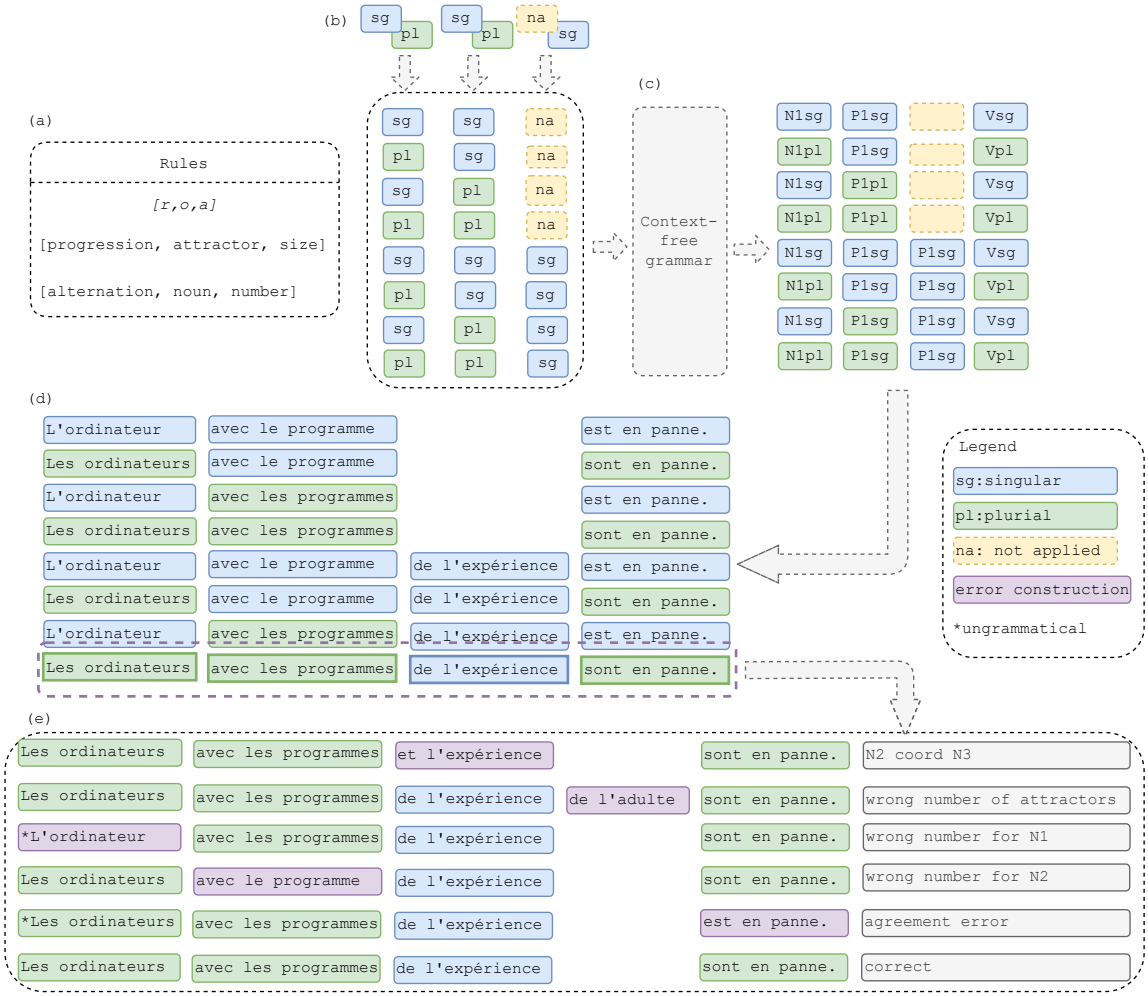
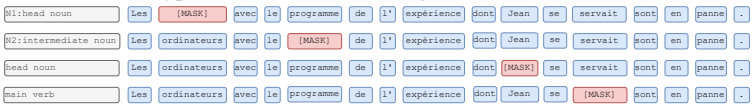


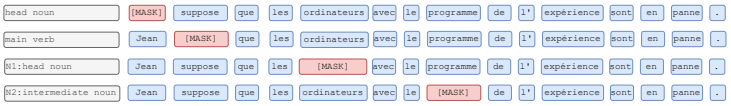
Figure 10:
Illustration of a BLMs problem generation process. Given sampled $[relation, object, attribute]$ rule tuples from (a), we first construct the abstract structure of the context with the values of the attributes of different objects in (b). We then derive, expand and prune the context-free grammar (with details in Figure 3) from each item's abstract structure into its corresponding sentence template in (c). In (d), we instantiate each item template into a sentence from the syntactic segment seed sets adapted from Franck et al. (2002). Finally, given the correct answer (last item), we modify one attribute at a time to obtain the relevant, minimally distinguishable and challenging candidate answer set in (e). The entire process is illustrated with an example of a progression in $[1, 2]$ constructed in a main clause. The error types across the dataset are uniformly distributed to avoid statistical bias.

A.2 Masking

(a) Type II Main clause masking



(b) Type II Relative clause masking



(c) Type II Completive clause masking

Figure 11: Creation of lexical variants for Type II: masking strategy.

A.3 Detailed results

TRAIN ON	TEST ON	FFNN	CNN
train on full training data			
type I	type I	**0.9957** (0)	0.9887 (0.0065)
	type II	**0.6508** (0)	0.6291 (0.0089)
	type III	**0.5724** (0)	0.4992 (0.0053)
type II	type I	**0.9870** (0)	0.9861 (0.0017)
	type II	**0.9578** (0)	0.9236 (0.0062)
	type III	**0.7469** (0)	0.6159 (0.0064)
type III	type I	**0.9740** (0)	0.8909 (0.0158)
	type II	**0.9055** (0)	0.7425 (0.0094)
	type III	**0.8792** (0)	0.6714 (0.0140)
train on the same amount of data (2073 instances: 1658 train/415 validation)			
type I	type I	**0.9896** (0.0035)	0.9827 (0)
	type II	**0.6491** (0.005)	**0.6492** (0)
	type III	**0.5644** (0.0038)	0.5370 (0)
type II	type I	0.5584 (0)	**0.6234** (0)
	type II	0.4779 (0)	**0.5229** (0)
	type III	0.4622 (0)	**0.4914** (0)
type III	type I	**0.5455** (0)	0.5368 (0)
	type II	0.4768 (0)	**0.4849** (0)
	type III	**0.4669** (0)	0.4664 (0)

Table 1: Average F1 (std) scores (to 4 decimal places) for the FFNN and CNN systems, over five runs. The highest value for each train/test combination is highlighted in bold.