Are there identifiable structural parts in the sentence embedding whole?

Anonymous ACL submission

Abstract

Sentence embeddings from transformer models encode in a fixed length vector much linguistic information. We explore the hypothesis that these embeddings consist of overlapping layers of information that can be separated, and on which specific types of information – such as information about chunks and their structural and semantic properties – can be detected. We show that this is the case using a dataset consisting of sentences with known chunk structure, and two linguistic intelligence datasets, solving which relies on detecting chunks and their grammatical number, and respectively, their semantic roles, and through analyses of the performance on the tasks and of the internal representations built during learning.

1 Introduction

Transformer architectures compress the information in a sentence – morphological, grammatical, semantic, pragmatic – into a one dimensional array of real numbers of fixed length. Sentence embeddings – usually fine-tuned – have proven useful for a variety of high-level language processing tasks (e.g. the GLUE tasks (Clark et al., 2020), story continuation (Ippolito et al., 2020)). Such higher-level tasks, however, might not necessarily require specific structural information. Sentence embeddings built using a BiLSTM model do seem to encode a range of information, from shallow (e.g. sentence length, word order) to syntactic (e.g. tree depth, top constituent) and semantic (e.g. tense, semantic mismatches) (Conneau et al., 2018). Investigation, or indeed, usage, of raw (i.e. not fine-tuned) sentence embeddings obtained from a transformer model are rare, possibly because most transformer models do not have a strong supervision signal on the sentence embedding. An investigation of the dimensions of BERT sentence embeddings using principal component analysis indicated that there is much correlation and redundancy, and that they encode more shallow information (length), rather than morphological, syntactic or semantic features (Nikolaev and Padó, 2023c). Moreover, analysis of information propagation through the model layers, and analysis of the sentence embeddings seem to show that much specialized information – e.g. POS, syntactic structure – while quite apparent at lower levels, gets lost towards the highest levels of the models (Rogers et al., 2020).

We hypothesize that different types of information are melded together, and no longer overtly accessible in the sentence embeddings. A raw sentence embedding – the encoding of the special [CLS]/$< s >$ token from the output of a pretrained transformer, not fine-tuned for a specific task – consists of overlapping layers[1] of information, similarly to an audio signal that is a combination of waves of different frequencies. The various types of information from the sentence – structural, semantic, etc. – are encoded on some of these layers. We use a convolutional neural network to separate different layers of information in a sentence embedding, and test whether syntactic and semantic structure – noun, verb and prepositional phrases, that may play different structural and semantic roles – can be identified on these layers.

Understanding what kind of information the sentence embeddings encode, and how, has multiple benefits: it connects internal changes in the model parameters and structure with changes in its outputs; it contributes to verifying the robustness of models and whether or not they rely on shallow or accidental regularities in the data; it narrows down the field of search when a language model produces wrong outputs, and it helps maximize the use of training data for developing more robust models from smaller textual resources.[2]

[1]Throughout this paper, by "layer" we mean "a stratum of information", not the layers of a transformer architecture.

[2]We will share the code and sentence data upon acceptance. The other datasets are publicly available.

2 Related work

How is the information from a textual input encoded by transformers? There are two main approaches to answer this question: (i) tracing specific information from input to output through the model's various layers and components, and (ii) investigating the generated embeddings. These investigations rely on probing the models, using purposefully built data that can implement different types of testing.

Tracing information through a transformer (Rogers et al., 2020) have shown that from the unstructured textual input, BERT (Devlin et al., 2019) is able to infer POS, structural, entity-related, syntactic and semantic information at successively higher layers of the architecture, mirroring the classical NLP pipeline (Tenney et al., 2019a). Further studies have shown that the information is not sharply separated, information from higher level can influence information at lower levels, such as POS in multilingual models (de Vries et al., 2020), or subject-verb agreement (Jawahar et al., 2019). Surface syntactic and semantic information seem to be distributed throughout BERT's layers (Niu et al., 2022; Nikolaev and Padó, 2023c). Attention is part of the process, as it helps encode various types of linguistic information (Rogers et al., 2020; Clark et al., 2019), syntactic dependencies (Htut et al., 2019), grammatical structure (Luo, 2021), and can contribute towards semantic role labeling (Tan et al., 2018; Strubell et al., 2018).

Word embeddings were shown to encode sentence-level information (Tenney et al., 2019b), including syntactic structure (Hewitt and Manning, 2019), even in multilingual models (Chi et al., 2020). Predicate embeddings contain information about its semantic roles structure (Conia and Navigli, 2022), embeddings of nouns encode subjecthood and objecthood (Papadimitriou et al., 2021). The averaged token embeddings are more commonly used as **sentence embeddings** (e.g. (Nikolaev and Padó, 2023a)), or the special token ([CLS]/<s>) embeddings are fine-tuned for specific tasks such as story continuation (Ippolito et al., 2020), sentence similarity (Reimers and Gurevych, 2019), alignment to semantic features (Opitz and Frank, 2022). This token averaging is justifiable as the learning signal for transformer models is stronger at the token level, with a much weaker objective at the sentence level – e.g. next sentence prediction (Devlin et al., 2018; Liu et al., 2019), sentence order prediction (Lan et al., 2019). Electra (Clark et al., 2020) does not either, but it relies on replaced token detection, which uses the sentence context to determine whether a (number of) token(s) in the given sentence were replaced by a generator sample. This training regime leads to sentence embeddings that perform well on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) and Stanford Question Answering (SQuAD) dataset (Rajpurkar et al., 2016), or detecting verb classes (Yi et al., 2022). Raw sentence embeddings also seemed to capture shallower information (Nikolaev and Padó, 2023c), but Nastase and Merlo (2023) show that raw sentence embeddings have internal structure that can encode grammatical sentence properties.

Probing models Analysis of BERT's inner workings has been done using probing classifiers (Belinkov, 2022), or through clustering based on the representations at the different levels (Jawahar et al., 2019). Probing has also been used to investigate the representations obtained from a pre-trained transformer model (Conneau et al., 2018). Elazar et al. (2021) propose amnesic probing to test both whether some information is encoded, and whether it is used. VAE-based methods (Kingma and Welling, 2013; Bowman et al., 2016) have been used to detect or separate specific information from input representations. Mercatali and Freitas (2021) capture discrete properties of sentences encoded with an LSTM (e.g. number and aspect of verbs) on the latent layer. Bao et al. (2019) and Chen et al. (2019) learn to disentangle syntactic and semantic information. Silva De Carvalho et al. (2023) learn to disentangle the semantic roles in natural language definitions from word embeddings.

Data For probing transfomers embeddings and behaviour, most approaches use datasets built by selecting, or constructing, sentences that exhibit specific structure and properties: definition sentences with annotated roles (Silva De Carvalho et al., 2023), sentences built according to a given template (Nikolaev and Padó, 2023b), sentences with specific structures for investigating different tasks, in particular SentEval (Conneau and Kiela, 2018) (Jawahar et al., 2019), example sentences from FrameNet (Conia and Navigli, 2022), a dataset with multi-level structure inspired by the Raven Progressive Matrices visual intelligence tests (An et al., 2023).

BLM agreement problem

Context Template			
NP-sg	PP1-sg		VP-sg
NP-pl	PP1-sg		VP-pl
NP-sg	PP1-pl		VP-sg
NP-pl	PP1-pl		VP-pl
NP-sg	PP1-sg	PP2-sg	VP-sg
NP-pl	PP1-sg	PP2-sg	VP-pl
NP-sg	PP1-pl	PP2-sg	VP-sg

Answer set	
NP-sg PP1-sg et NP2 VP-sg	Coord
NP-pl PP1-pl NP2-sg VP-pl	correct
NP-sg PP1-sg VP-sg	WNA
NP-pl PP1-pl NP2-pl VP-sg	AE_V
NP-pl PP1-sg NP2-pl VP-sg	AE_N1
NP-pl PP1-pl NP2-sg VP-sg	AE_N2
NP-pl PP1-sg PP1-sg VP-pl	WN1
NP-pl PP1-pl PP2-pl VP-pl	WN2

BLM verb alternation problem

Context Template			
NP-Agent	Verb	NP-Loc	PP-Theme
NP-Theme	VerbPass	PP-Agent	
NP-Theme	VerbPass	PP-Loc	PP-Agent
NP-Theme	VerbPass	PP-Loc	
NP-Loc	VerbPass	PP-Agent	
NP-Loc	VerbPass	PP-Theme	PP-Agent
NP-Loc	VerbPass	PP-Theme	

Answer set	
NP-Agent Verb NP-Theme PP-Loc	Correct
NP-Agent *VerbPass NP-Theme PP-Loc	AgentAct
NP-Agent Verb NP-Theme *NP-Loc	Alt1
NP-Agent Verb *PP-Theme PP-Loc	Alt2
NP-Agent Verb *[NP-Theme PP-Loc]	NoEmb
NP-Agent Verb NP-Theme *PP-Loc	LexPrep
NP-Theme Verb NP-Agent PP-Loc	SSM1
NP-Loc Verb NP-Agent PP-Theme	SSM2
NP-Theme Verb NP-Loc PP-Agent	AASSM

Figure 1: Structure of two BLM problems, in terms of chunks in sentences and sequence structure.

3 Data

Our main object of investigation are chunks, sequence of adjacent words that segment a sentence (as defined initially in (Abney, 1992), (Collins, 1997) and then (Tjong Kim Sang and Buchholz, 2000). To investigate whether chunks and their properties are identifiable in sentence embeddings, we use two types of data: (i) sentences with known chunk pattern, described in Section 3.1; (ii) two datasets with multi-level structure built for linguistic intelligence tests for language models (Merlo, 2023), described in Section 3.2.

3.1 Sentences

Sentences are built from a seed file containing noun, verb and prepositional phrases, including singular/plural variations. From these chunks, we built sentences with all (grammatically correct) combinations of `np (pp1 (pp2)) vp`[3]. For each chunk pattern p of the 14 possibilities (for instance, p = "np-s pp1-s vp-s"), all corresponding sentences are collected into a set S_p.

We generate an instance for each sentence s from the sets S_p as a triple (in, out^+, out^-), where $in = s$ is the input, out^+ is the correct output, which is a sentence different from s but having the same chunk pattern. out^- are N_{negs} incorrect outputs, randomly chosen from the sentences that have a chunk pattern different from s. The algorithm for building the data and a sample line and generated sentences are shown in appendix A.1.

[3]We use BNF notation: pp_1 and pp_2 may be included or not, pp_2 may be included only if pp1 is included

From the generated instances, we sample uniformly, based on the pattern of the input sentence, approximately 4000 instances, randomly split 80:20 into train:test. The train part is further split 80:20 into train:dev, resulting in a 2576:630:798 split for train:dev:test. We use a French and an English seed file and generate French and English variations of the dataset, with the same statistics.

3.2 Blackbird Language Matrices

Blackbird Language Matrices (BLMs) (Merlo, 2023) are language versions of the visual Raven Progressive Matrices. They are multiple-choice problems, where the input is a sequence of sentences built using specific rules, and the answer set consists of a correct answer that continues the input sequence, and several incorrect options that are built by corrupting some of the underlying generating rules of the sentences in the input sequence. In a BLM matrix, all sentences share a targeted linguistic phenomenon, but differ in other aspects relevant for the phenomenon in question. Thus, BLMs, like their visual counterpart RPMs, require identifying the entities (the chunks), their relevant attributes (their morphological or semantic properties) and their connecting operators, to find the underlying rules that guide to the correct answer.

We use two BLM datasets, which encode two different linguistic phenomena, each in a different language: (i) BLM-AgrF – subject verb agreement in French (An et al., 2023), and (ii) BLM-s/lE – verb alternations in English (Samo et al., 2023). The structure of these datasets – in terms of the sentence chunks and sequence structure – is shown

	Subj.-verb agr.	Verb alternations	
		ALT-ATL	ATL-ALT
Type I	2000:252	2000:375	2000:375
Type II	2000:4866	2000:1500	2000:1500
Type III	2000:4869	2000:1500	2000:1500

Table 1: Train:Test statistics for the two BLM problems.

in Figure 1, and concrete examples are shown in appendices A.2, A.3.

BLM datasets also have a lexical variation dimension. There are three variants: type I – minimal lexical variation for sentences within an instance, type II – one word difference across the sentences within an instance, type III – maximal lexical variation within an instance. This allows for investigations in the impact of lexical variation on learning the relevant structures to solve the problems.

We use the BLM-s/lE dataset as is. We built a variation of the BLM-AgrF (An et al., 2023) that separates clearly sequence-based errors (WN1 and WN2 in the agreement scheme presented in Figure 1) from other types of errors. We include erroneous answers that have correct agreement, but do not respect the pattern of the sequence, to be able to contrast linguistic errors from errors in identifying sentence parts.

Datasets statistics Table 1 shows the datasets statistics for the BLM problems. After splitting each subset 90:10 into train:test subsets, we randomly sample 2000 instances as train data. 20% of the train data is used for development. Types I, II, III correspond to different amounts of lexical variation within a problem instance.

4 Experiments

We aim to determine whether specific kinds of sentence parts – chunks – are identifiable in transformer-based sentence embeddings. We approach this problem from two angles. First, using sentences and a VAE-based system, we test whether we can compress sentences into a smaller representation on the latent layer that captures information about the chunk structure of the sentence (Section 4.1 below). Second, to see if the chunks thus identified are being used in a separate task, we combine the compression of the sentence representation with the BLM problems, where a crucial part of the solution lies in identifying the structures of sentences and their sequence in the input (Section 4.2 below).

As sentence representations, we use the embeddings of the $< s >$ character read from the last layer of the Electra (Clark et al., 2020) pretrained model[4].

4.1 Parts in sentences

We test whether sentence embeddings contain information about the chunk structure of the corresponding sentences by compressing them into a lower dimensional representation in a VAE-like system.

4.1.1 Experimental set-up

The architecture of the sentence-level VAE is similar to a previously proposed system (Nastase and Merlo, 2023). The encoder consists of a CNN layer with a 15x15 kernel, which is applied to a 32x24-shaped sentence embedding,[5] followed by a linear layer that compresses the output of the CNN into a latent layer of size 5. The decoder is a mirror-image of the encoder, and unpacks a sampled latent vector into a 32x24 sentence representation.

An instance consists of a triple (in, out^+, out^-), where in is an input sentence with embedding e_i and chunk structure p, out^+ is a sentence with embedding e_j with same chunk structure p, and out^- is a set of N_{negs} sentences with embeddings e_k, each of which has a chunk pattern different from p (and different from each other). The input e_i is encoded into a latent representation z_i, from which we sample a vector $\tilde{z}_i$, which is decoded into the output $\hat{e}_i$. We enforce that the latent encodes the structure of the input sentence by using a max-margin loss function. This loss function assigns a higher score to e_j than to e_k, relative to $\hat{e}_i$. Recall that e_j has the same chunk structure as the input $\hat{e}_i$.

$$loss(e_i) = \\ maxmargin(\hat{e}_i, e_j, e_k) + KL(z_i||\mathcal{N}(0,1))$$

$$maxmargin(\hat{e}_i, e_j, e_k) = \\ max(0, 1 - score(\hat{e}_i, e_j) + \frac{\sum_{k=1}^{N_{negs}} score(\hat{e}_i, e_k)}{N_{negs}})$$

The *score* between two embeddings is the cosine similarity. At prediction time, the sentence from the $\{out^+\} \cup out^-$ options that has the highest score relative to the input sentence is taken as the correct answer.

4.1.2 Analysis

To assess whether the correct patterns of chunks are detected in sentences, we analyze the results for

[4]Electra pretrained model: *google/electra-base-discriminator*

[5]Nastase and Merlo (2023) show that task-relevant information is more easily accessible in transformer-based sentence embeddings reshaped as two-dimensional arrays.

the experiments described in the previous section in two ways: (i) analyze the output of the system, in terms of average F1 score over three runs and confusion matrices; (ii) analyze the latent layer, to determine whether chunk patterns are encoded in the latent vectors (for instance, latent vectors cluster according to the pattern of their corresponding sentences).

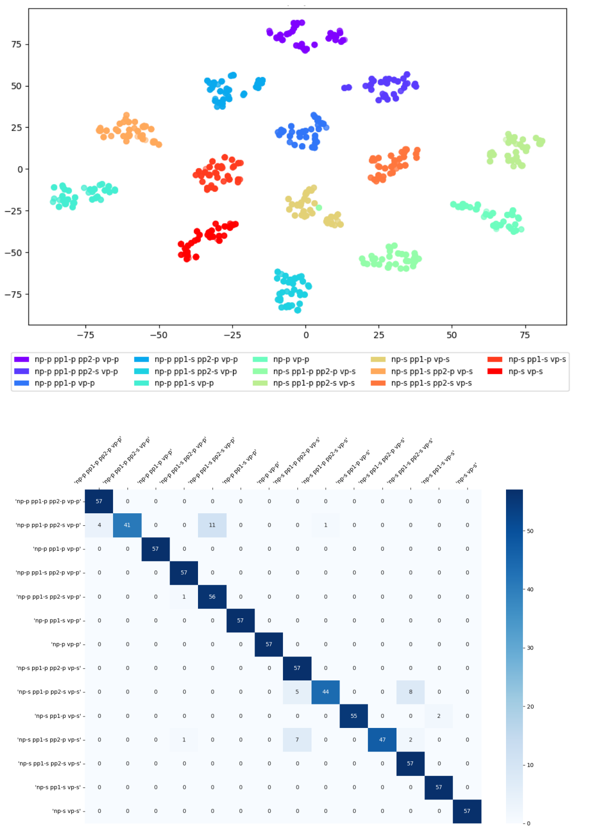


Figure 2: Chunk identification results: tSNE projections of the latent vectors for the French dataset, and confusion matrix of the system output. The results for English are similar.

If we consider the multiple choice task as a binary task (Has the system built a sentence representation that is closest to the correct answer?), the system achieves an average positive class F1 score (and standard deviation) over three runs of 0.9992 (0.01) for the French dataset, and 0.997 (0.0035) for the English dataset. For added insight, for one trained model for each of the French and English data, we compute a confusion matrix, based on the pattern information for out^+, out^-. The results for French are presented in Figure 2.

To check whether chunk information is present in the latent layer, we plot the projection in two dimensions of the latent vectors. The plot shows a very crisp clustering of latents that correspond to input sentences with the same chunk pattern, despite the fact that some patterns differ by only one attribute (the grammatical number) of one chunk.

To understand how chunk information is encoded on the latent layer we perform latent traversals: for each instance in the test data, we modify the value of each unit in the latent layer with ten values in the min-max range of that unit, based on the training data. A sample of confusion matrices with interventions on the latent layer is shown in Figure 3.

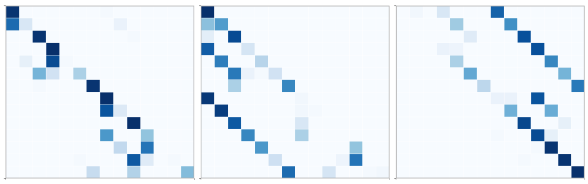


Figure 3: The impact on reconstructing sentences with the same pattern when modifying the latent layer with values in their respective min-max range (based on the training data) – sample confusion matrices.

The confusion matrices presented as heatmaps in Figure 3 (and a larger version with labels in Figure 10 in Appendix A.5) show that specific changes to the latent vectors decrease the differentiation among patterns, as expected if chunk pattern information were encoded in the latent vectors. Changes to latent 1 cause patterns that differ in the grammatical number of *pp*2 not to be distinguishable (left matrix). Changes to latent units 2 and 3 lead to the matrices 2 and 3 in the figure, where patterns that have different subject-verb grammatical number to become indistinguishable.

4.2 Parts in sentences for BLM tasks

The first experiment shows that compressing sentence representations results in latent vectors containing chunk information. To test if these latent representations also contain information about chunk properties relevant to a task, we solve the BLM task.

4.2.1 Experimental set-up

To explore how chunk information in the sentence embeddings is used in a task, we solve the BLM problems. The BLM problems encode a linguistic phenomenon in a sequence of sentences that have regular and relevant structure, which serves to emphasize and reinforce the encoded phenomenon. BLMs are inspired by Raven Progressive Matrices, whose solution has been shown to require solving two main subtasks: identifying the objects and object attributes that occur in the visual frames, and decomposing the main problem into subproblems, based on object and attribute identification, in a way that allows detecting the global pattern or underlying rules. It has also been shown that being able to solve RPMs requires being able to handle item novelty (Carpenter et al., 1990). We model these ingredients of the solution of a RPM/BLM explicitly by using the two-level intertwined architecture illustrated in Figure 4 – one level for detecting sentence structure, one for detecting the correct answer based on the sequence of structures and the targeted grammatical phenomenon. Item novelty is modeled through the three levels of lexicalisation (section 3).

The sentence level is essentially the system described above. The representation on the latent layer is used to represent each of the sentences in the input sequence, and to solve the problem at the task level. The two layers are trained together.

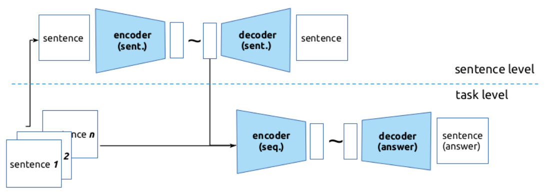


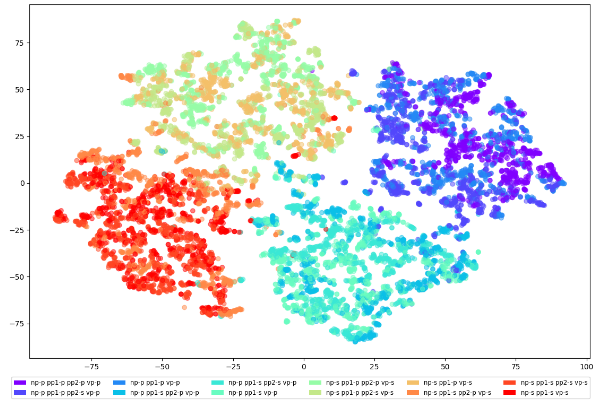
Figure 4: A two-level VAE: the sentence level learns to compress a sentence into a representation useful to solve the BLM problem on the task level.

An instance for a BLM problem consists of an ordered sequence S of sentences, $S = \{s_i | i = 1, 7\}$ as input, and an answer set A with one correct answer a_c, and several incorrect answers a_{err}. The sentences in S are passed as input to the sentence-level VAE. The sampled latent representations from this VAE are used as the representations of the sentences in S. These representations are passed as input to the BLM-level VAE, in the same order as S. An instance for the sentence-level VAE consists of a triple (in, out^+, out^-). For our two-level system, we must construct this triple from the input BLM instance: $in \in S$, $out^+ = in$, and $out^- = \{s_k | s_k \in S, s_k \neq in\}$.

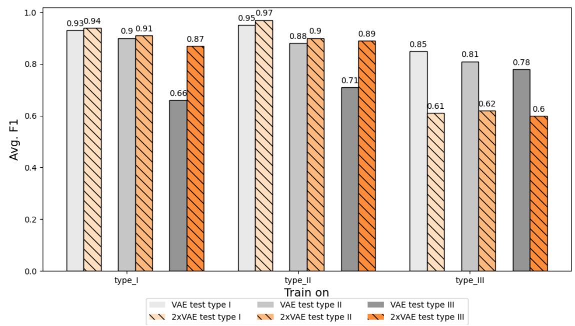
The loss combines the loss signal from the two levels:

$$loss = \\ maxmargin_{sent} + KL_{sent} + \\ maxmargin_{task} + KL_{seq}$$

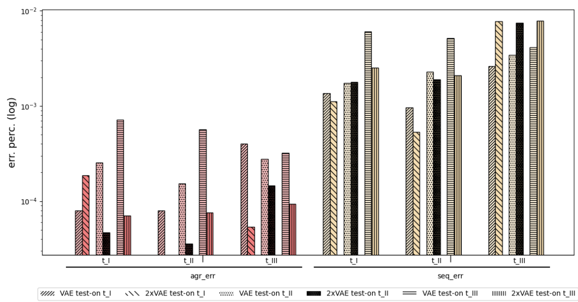
The $maxmargin$ and the scoring of the reconstructed sentence at the sentence level, and the constructed answer at the task level are computed as described in Section 4.1.



TSNE projection of latent representations from the latent layer of the sentence level for the sentences in BLM contexts in the training data, coloured by the chunk pattern.



Average F1 score over 3 runs, grouped by training data on the x-axis, tested on type I, II, III in different shades.



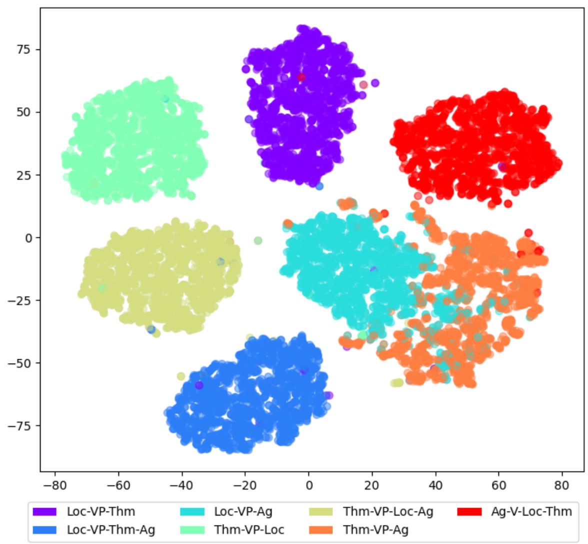
Sequence vs. agreement errors analysis.

Figure 5: VAE vs 2-level VAE (2xVAE) on the agreement BLM problem

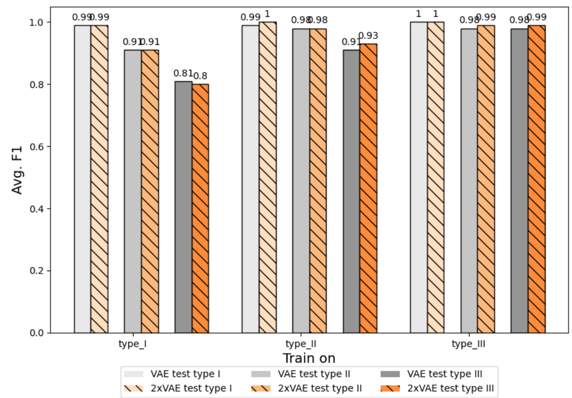
We run experiments on the BLMs for agreement and for verb alternation. While the information necessary to solve the agreement task is more structural, solving the verb alternation task requires not only structural information concerning chunks, but also semantic information, as syntactically similar chunks play different roles in a sentence.

4.2.2 Analysis

The results show that this organisation of the system leads to better results compared to the one-level process for these structure-based linguistic problems, thereby providing additional support to

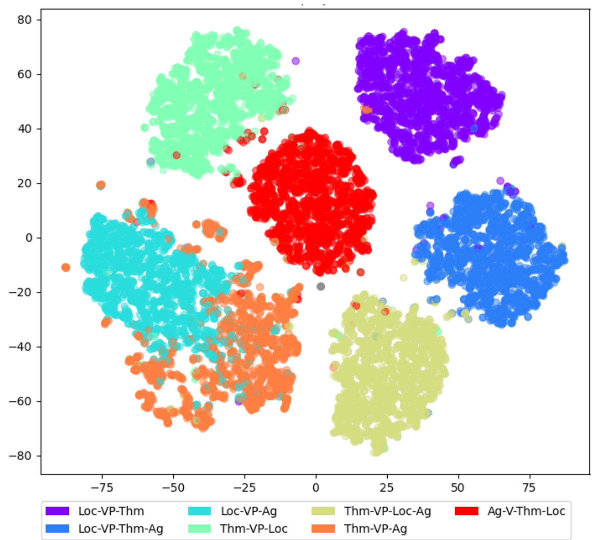


TSNE projection of latent representations from the latent layer of the sentence level for the sentences in BLM contexts in the training data, coloured by the pattern of semantic roles.

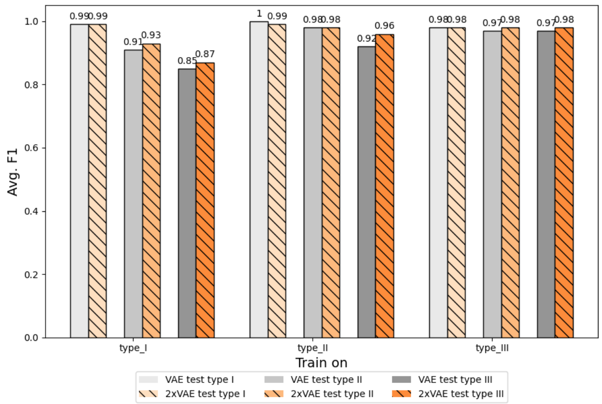


Average F1 score over 3 runs

Figure 6: VAE vs 2-level VAE (2xVAE) on the verb alternation BLM problem, Group 1



TSNE projection of latent representations from the latent layer of the sentence level for the sentences in BLM contexts in the training data, coloured by the pattern of semantic roles.



Average F1 score over 3 runs

Figure 7: VAE vs 2-level VAE (2xVAE) on the verb alternation BLM problem, Group 2

our hypothesis that chunks and their attributes are detectable in sentence embeddings.

We provide results in terms of F1 scores on the task, and analysis of the representations on the latent layer of the sentence level of the system.

Figure 5 shows the results on the BLM agreement task and the error analysis (detailed results are in the appendix). The results on the task (left panel) provide several insights. First, from the latent representation analysis, we note that while the sentence representations on the latent layer are not as crisply separated by their chunk pattern as for the experiment in Section 4.1, there is a clear separation in terms of the grammatical number of the subject and the verb. This is not surprising as the focus of the task is subject-verb agreement. However, as the further results in term of F1 and error analysis on the task show, there is enough information in these compressed latent representation to capture the structural regularities imposed by the patterns of chunks in the input sequence.

Second, from the results in terms of F1, we note that the two-level process generalizes better from simpler data – learning on type I and type II leads to better results on all test data, with the highest improvement when tested on type III data, which has the highest lexical variation. Furthermore, the two-level models learned when training on the lexically simpler data perform better when tested on the type III data than the models learned on type III data itself. This result not only indicates that structure information is more easily detectable when lexical variation is less of a factor, but more importantly, that chunk information is separable from other types of information in the sentence embedding, as the patterns detecting it can be applied successfully for data with additional (lexical) variation.[6]

Further confirmation of the fact that the sentence level learns to compress sentences into a latent that captures structural information comes from the error analysis, shown in the bottom panel of Figure 5. Lower rate of sequence errors, which are correct from the point of view of the targeted phenomenon – as described in section 3.2 – indicate that there is structure information in the compressed sentence latents.

It is possible that the one-level VAE also detects chunk information in the input sequence, given the high performance on the task. But the fact that the one-level model makes more sequence-based errors indicates that modeling structural information separately is not only possible, but also beneficial for some tasks.

The results on the verb alternation BLMs are shown in Figures 6 and 7. In this problem, and unlike the verb-agreement BLM task, structurally similar chunks - NPs, PPs – play different semantic roles in the verb alternation data, as shown in Figure 1. Other attributes of chunks that are relevant to the current problem – in this case, semantic roles – are separated from the sentence embedding whole. This is apparent not only through the F1 results on the task, but also, and maybe more clearly, from the projection of the latent representations from the sentence level, where the separation of the different chunk syntactic and semantic patterns is clear for both groups. For both data subsets, the closest representations are two that have the same syntactic pattern: *NP VerbPass PP*, but semantically differ: *NP-Theme VerbPass PP-Agent* vs. *NP-Loc VerbPass PP-Agent*.

4.3 Discussion

We performed two types of experiments: (i) using individual sentences, and an indirect supervision signal about the sentence structure, (ii) incorporating a sentence representation compression step in a task-specific setting. We used two tasks, one which relies on more structural information (subject-verb agreement), and one that also relies on semantic information about the chunks (verb alternation).

We have investigated each set-up in terms of the results on the task – as average F1 scores, and through error analysis – and in terms of internal representations on the latent layer of an encoder-decoder architecture.

This dual analysis allows us to conclude not only that a task is solved correctly, but that it is solved using structural, morphological and semantic information from the sentence. We found that information about (varying numbers of) chunks – noun, verb and prepositional phrases – and their task-relevant attributes, whether morphological or semantic, can be detected in sentence embeddings from a pretrained transformer model.

5 Conclusions

Sentence embeddings obtained from transformer models are compact representations, compressing much knowledge – morphological, grammatical, semantic, pragmatic –, expressed in text fragments of various length, into a vector of real numbers of fixed length. If we view the sentence embedding as overlapping layers of information, in a manner similar to audio signals which consist of overlapping signals of different frequencies, we can distinguish specific information among these layers. In particular, we have shown that we can detect information about chunks – noun/verb/prepositional phrases – and their task-relevant attributes in these compact sentence representations.

These building blocks can be further used in lexically-novel instances to solve tasks that require analytical reasoning, demonstrating that solutions to this task are achieved through abstract steps typical of fluid intelligence.

6 Limitations

We have performed experiments on datasets containing sentences with specific structure and properties to be able to determine whether the type of information we targeted can be detected in sentence embeddings. We applied our framework on a particular pretrained transformer model – Electra – which we chose because of the stronger influence of the full context on producing sentence embeddings. Different transformer models may produce

[6]It might appear surprising that the two-level approach leads to lower performance on type III data, particularly when lexical variation had not been an issue for the sentence representation analysis (Section 4.1). The difference comes from the way the instances were formed, on the fly, for the two-level process: the positive sentence to be reconstructed is the same as the input, instead of being a sentence that has the same structure, but different lexical material. This is because all sentences in the sequence have different structures. We think this weakens the (indirect) supervision signal – as the correct answer is distinct from the other options. This is not the case for type I and II data, where, because of the very similar lexical material, the distinction between the correct and incorrect answers reduce to the structure. We plan to confirm this in future work using a pre-trained sentence-level VAE.

different encoding patterns in the sentence embeddings.

References

Steven Abney. 1992. Prosodic structure, performance structure and phrase structure. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Aixiu An, Chunyang Jiang, Maria A. Rodriguez, Vivi Nastase, and Paola Merlo. 2023. BLM-AgrF: A new French benchmark to investigate generalization of agreement in neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1363–1374, Dubrovnik, Croatia. Association for Computational Linguistics.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Patricia A Carpenter, Marcel A Just, and Peter Shell. 1990. What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test. *Psychological review*, 97(3):404.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence 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 2453–2464, Minneapolis, Minnesota. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Simone Conia and Roberto Navigli. 2022. Probing for predicate argument structures in pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4622–4632, Dublin, Ireland. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about BERT's layers? a closer look at the NLP pipeline in monolingual and multilingual models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

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.

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.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in bert track syntactic dependencies? *Preprint*, arXiv:1911.12246.

Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. 2020. Toward better storylines with sentence-level language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7472–7478, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ziyang Luo. 2021. Have attention heads in BERT learned constituency grammar? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 8–15, Online. Association for Computational Linguistics.

Giangiacomo Mercatali and André Freitas. 2021. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3547–3556, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paola Merlo. 2023. Blackbird language matrices (BLM), a new task for rule-like generalization in neural networks: Motivations and formal specifications. *ArXiv*, cs.CL 2306.11444.

Vivi Nastase and Paola Merlo. 2023. Grammatical information in BERT sentence embeddings as two-dimensional arrays. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 22–39, Toronto, Canada. Association for Computational Linguistics.

Dmitry Nikolaev and Sebastian Padó. 2023a. Investigating semantic subspaces of transformer sentence embeddings through linear structural probing. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 142–154, Singapore. Association for Computational Linguistics.

Dmitry Nikolaev and Sebastian Padó. 2023b. Representation biases in sentence transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3701–3716, Dubrovnik, Croatia. Association for Computational Linguistics.

Dmitry Nikolaev and Sebastian Padó. 2023c. The universe of utterances according to BERT. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 99–105, Nancy, France. Association for Computational Linguistics.

Jingcheng Niu, Wenjie Lu, and Gerald Penn. 2022. Does BERT rediscover a classical NLP pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Juri Opitz and Anette Frank. 2022. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 625–638, Online only. Association for Computational Linguistics.

Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. 2021. Deep subjecthood: Higher-order grammatical features in multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Giuseppe Samo, Vivi Nastase, Chunyang Jiang, and Paola Merlo. 2023. BLM-s/lE: A structured dataset of English spray-load verb alternations for testing generalization in LLMs. In *Findings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Danilo Silva De Carvalho, Giangiacomo Mercatali, Yingji Zhang, and André Freitas. 2023. Learning disentangled representations for natural language definitions. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1371–1384, Dubrovnik, Croatia. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *The Seventh International Conference on Learning Representations (ICLR)*, pages 235–249.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

David Yi, James Bruno, Jiayu Han, Peter Zukerman, and Shane Steinert-Threlkeld. 2022. Probing for understanding of English verb classes and alternations in large pre-trained language models. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 142–152, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

A Appendix

A.1 Sentence data

To build the sentence data, we use a seed file that was used to generate the subject-verb agreement data. A seed, consisting of noun, prepositional and verb phrases with different grammatical numbers, can be combined to build sentences consisting of different sequences of such chunks. Table 2 includes a partial line from the seed file, from which individual sentences and a BLM instance can be constructed. We use French and English versions of the seed file to build the corresponding datasets.

Subj_sg	Subj_pl	P1_sg	P1_pl	P2_sg	P2_pl	V_sg	V_pl
The computer	The computers	with the program	with the programs	of the experiment	of the experiments	is broken	are broken

Sent. with different chunks	
The computer is broken.	np-s vp-s
The computers are broken.	np-p vp-p
The computer with the program is broken.	np-s pp1-s vp-s
...	...
The computers with the programs of the experiments are broken.	np-p pp1-p pp2-p vp-p

a BLM instance
Context:
The computer with the program is broken.
The computers with the program are broken.
The computer with the programs is broken.
The computers with the programs are broken.
The computer with the program of the experiment is broken.
The computers with the program of the experiment are broken.
The computer with the programs of the experiment is broken.
Answer set:
The computers with the programs of the experiment are broken.
The computers with the programs of the experiments are broken.
The computers with the program of the experiment are broken.
The computers with the program of the experiment is broken.
...

Table 2: A line from the seed file on top, and a set of individual sentences built from it, as well as one BLM instance.

The algorithm to produce a dataset from the generated sentences is detailed in Figure 8 below.

Data = []; N_{negs}
for patterns p **do**
  for $s_i \in S_p$ **do**
    in = s_i
    for $s_j \in S_p$ **do**
      out$^+$ = s_j
      out$^-$ = $\{s_k, k \in range(N_{negs}), s_k \in S_{\neg p}\}$
      Data = Data $\cup$ [(in, out$^+$, out$^-$)]
    end for
  end for
end for

Figure 8: Data generation algorithm

A.2 Example of data for the agreement BLM

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

Table 3: Examples from (Franck et al., 2002), manually completed and expanded sentences based on these examples, and seeds made based on these sentences for subject-verb agreement BLM 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 le program de l'experiénce sont en panne.	agreement error
5	Les ordinateurs avec les programmes de l'experiénce sont en panne.	wrong nr. for 1^{st} attractor noun
6	Les ordinateurs avec les programmes de les experiénces sont en panne.	wrong nr. for 2^{nd} attractor noun

Table 4: BLM instances for verb-subject agreement, with 2 attractors (programme, experiénce), and three clause structures. And candidate answer set for a problem constructed from lines 1-7 of the main clause sequence.

A.3 Example of data for the verb alternation BLM

TYPE I

EXAMPLE OF CONTEXT
The buyer can load the tools in bags.
The tools were loaded by the buyer
The tools were loaded in bags by the buyer
The tools were loaded in bags
Bags were loaded by the buyer
Bags were loaded with the tools by the buyer
Bags were loaded with the tools
???

EXAMPLE OF ANSWERS
The buyer can load bags with the tools
The buyer was loaded bags with the tools
The buyer can load bags the tools
The buyer can load in bags with the tools
The buyer can load bags on sale
The buyer can load bags under the tools
Bags can load the buyer with the tools
The tools can load the buyer in bags
Bags can load the tools in the buyer

Figure 9: Example of Type I context sentences and answer set.

A.4 Experimental details

All systems used a learning rate of 0.001 and Adam optimizer, and batch size 100. The system was trained for 300 epochs for all experiments.

The experiments were run on an HP PAIR Workstation Z4 G4 MT, with an Intel Xeon W-2255 processor, 64G RAM, and a MSI GeForce RTX 3090 VENTUS 3X OC 24G GDDR6X GPU.

The **sentence-level encoder decoder** has 106 603 parameters. It consists of an encoder with a CNN layer followed by a FFNN layer. The CNN input has shape 32x24. We use a kernel size 15x15 with stride 1x1, and 40 channels. The linearized CNN output has 240 units, which the FFNN compresses into the latent layer of size 5+5 (mean+std). The decoder is a mirror of the encoder, which expands a sampled latent of size 5 into a 32x24 representation.

The **two-level system** consists of the sentence level encoder-decoder described above, and a task-specific layer. The input to the task layer is a 7x5 input (sequence of 7 sentences, whose representation we obtain from the latent of the sentence level), which is compressed using a CNN with kernel 4x4 and stride 1x1 and 32 channels into ... units, which are compressed using a FFNN layer into a latent layer of size 5+5 (mean+std). The decoder consists of a FFNN which expands the sampled latent of size 5 into 7200 units, which are then processed through a CNN with kernel size 15x15 and stride 1x1, and produces a sentence embedding of size 32x24. The two level system has 178 126 parameters.

A.5 Sentence-level analysis

A.5.1 Sample confusion matrices for altered latent values

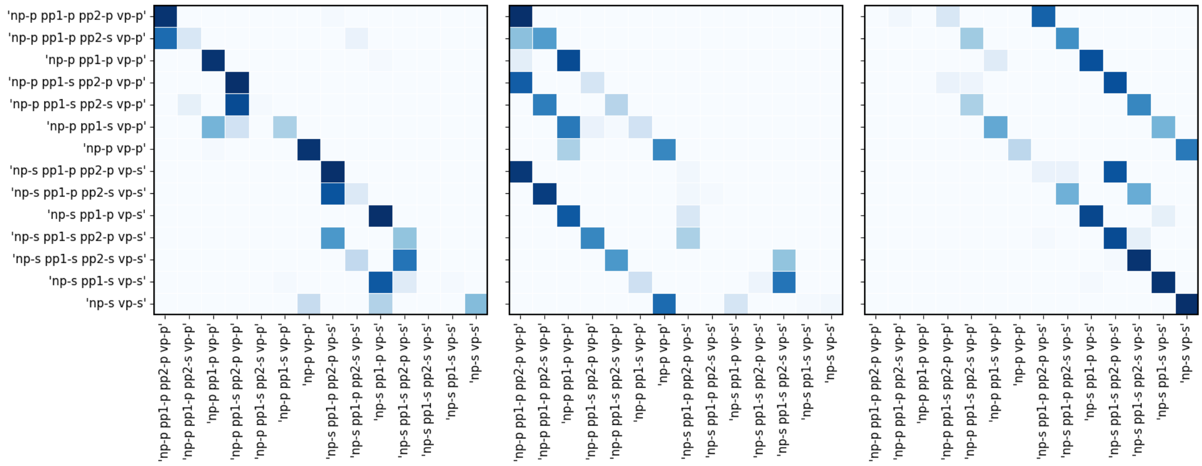


Figure 10: Confusion matrices for altered values on units 1 (left matrix), unit 2 (middle matrix) and unit 3 (right matrix)

Each matrix shows a particular way of conflating different patterns:

- changes to values in unit 1 of the latent lead to patterns that differ in the grammatical number of *pp*2 to become indistinguishable
- changes to values in units 2 and 3 of the latent lead to the conflation of patterns that have different subject-verb numbers.

A.5.2 Sentence-level analysis for English data

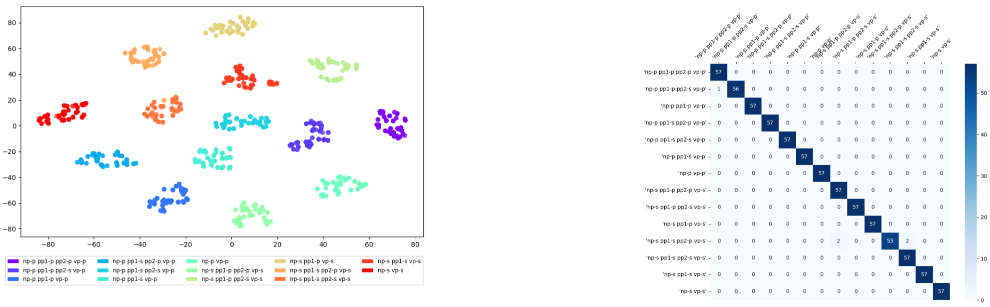


Figure 11: Chunk identification results: tSNE projections of the latent vectors for the English dataset, and confusion matrix of the system output.

A.6 Detailed task results

TRAIN ON	TEST ON	VAE	2 LEVEL VAE
BLM agreement			
type_I	type_I	0.929 (0)	**0.935** (0.0049)
type_I	type_II	0.899 (0)	**0.908** (0.0059)
type_I	type_III	0.662 (0)	**0.871** (0.0092)
type_II	type_I	0.948 (<e-10)	**0.974** (0.0049)
type_II	type_II	0.879 (<e-10)	**0.904** (0.0021)
type_II	type_III	0.713 (0)	**0.891** (0.0015)
type_III	type_I	**0.851** (0.037)	0.611 (0.1268)
type_III	type_II	**0.815** (0.0308)	0.620 (0.1304)
type_III	type_III	**0.779** (0.0285)	0.602 (0.1195)
BLM verb alternation group 1			
type_I	type_I	0.989 (0)	**0.995** (<e-10)
type_I	type_II	0.907 (0)	**0.912** (0.0141)
type_I	type_III	**0.809** (0)	0.804 (0.0167)
type_II	type_I	0.989 (0)	**0.996** (0.0013)
type_II	type_II	0.979 (<e-10)	**0.984** (0.0016)
type_II	type_III	0.915 (0)	**0.928** (0.0178)
type_III	type_I	0.997 (0)	**0.999** (0.0013)
type_III	type_II	0.977 (0)	**0.986** (0.0027)
type_III	type_III	0.98 (0)	**0.989** (0.0003)
BLM verb alternation group 2			
type_I	type_I	**0.992** (0)	0.987 (0.0033)
type_I	type_II	0.911 (0)	**0.931** (0.0065)
type_I	type_III	0.847 (0)	**0.869** (0.0102)
type_II	type_I	**0.997** (0)	0.993 (0.0025)
type_II	type_II	**0.978** (<e-10)	**0.978** (0.0017)
type_II	type_III	0.923 (0)	**0.956** (0.0023)
type_III	type_I	0.979 (<e-10)	**0.981** (0.0022)
type_III	type_II	0.972 (0)	**0.975** (0.0005)
type_III	type_III	0.967 (0)	**0.977** (0.0022)

Table 5: Analysis of systems: average F1 (std) scores (over 3 runs) for the VAE and 2xVAE systems. The highest value for each train/test combination highlighted in bold.

A.7 Detailed error results

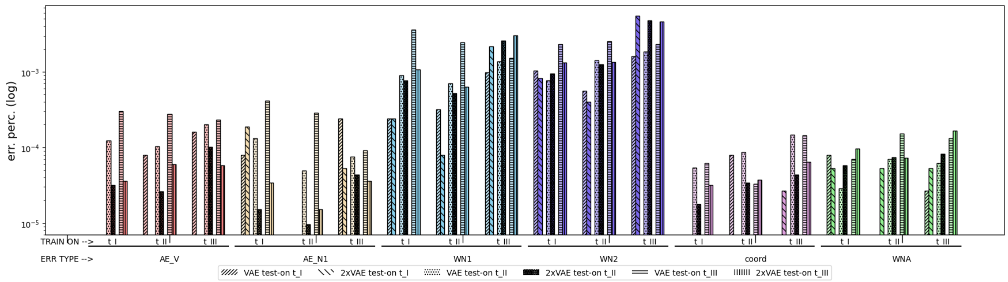


Figure 12: Agreement error analysis: y-axis is the log of error percentages. N1_alter and N2_alter are sequence errors.

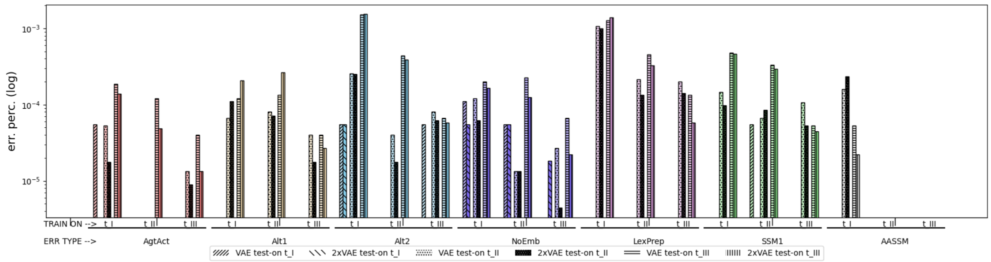


Figure 13: Verb alternation group1 error analysis: y-axis is the log of error percentages.

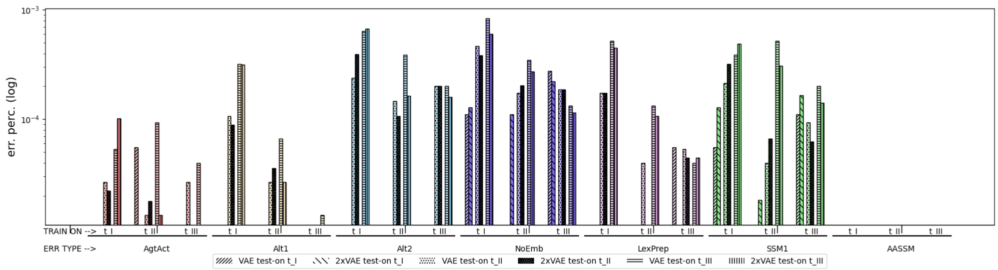


Figure 14: Verb alternation group2 error analysis: y-axis is the log of error percentages.