Modeling Construction Grammar's Way into NLP: Insights from negative results in automatically identifying schematic clausal constructions in Brazilian Portuguese

Arthur Lorenzi[1], Vânia Gomes de Almeida[1], Ely Edison Matos[1], Tiago Timponi Torrent[1,2]
[1] FrameNet Brasil Lab, Graduate Program in Linguistics, Federal University of Juiz de Fora
[2] Brazilian National Council for Scientific and Technological Development – CNPq
{arthur.lorenzi,vania.almeida}@estudante.ufjf.br
{tiago.torrent,ely.matos}@ufjf.br

Abstract

This paper reports on negative results in a task of automatic identification of schematic clausal constructions and their elements in Brazilian Portuguese. The experiment was set up so as to test whether form and meaning properties of constructions, modeled in terms of Universal Dependencies and FrameNet Frames in a Constructicon, would improve the performance of transformer models in the task. Qualitative analysis of the results indicate that alternatives to the linearization of those properties, dataset size and a post-processing module should be explored in the future as a means to make use of information in Constructicons for NLP tasks.

1 Introduction

Constructional approaches to language description can be traced back to early work by Fillmore (1968), which later gave rise to a myriad of approaches sharing the common assumptions that (a) constructions are learned pairings of form and function related to one another in a network, and (b) grammar does not rely on transformations and derivation, instead it is directly associated with function (Goldberg, 2013).

From the 2000's on, computational implementations of Construction Grammar started being built both in terms of language resources comprising of collections of constructions called *Constructicons* (Fillmore, 2008; Lyngfelt et al., 2012; Ohara, 2014; Torrent et al., 2014; Ziem and Boas, 2017), and proofs of concept, namely constructional parsers (Bryant, 2008; Matos et al., 2017).

As a natural consequence of the focus of constructionist analysis on families of constructions, Constructicons typically start by modeling the same kind of phenomena, leaving more schematic and foundational language structures, clausal and phrasal constructions, respectively, for later. These kinds of constructions represent a challenge for both Constructicography, that is, the process of describing and modeling constructions in a resource (Lyngfelt et al., 2018), and for constructional parsing, since schematic clausal constructions, as opposed to idioms, are typically difficult or impossible to describe in terms of the presence of distinctive lexical fillers. Moreover, it is common for those constructions to share constituency properties. As an example, consider (1) and (2), both sentences share the same syntactic structure in Brazilian Portuguese, but express opposite types of semantic events (controlled × uncontrolled activity), thus representing instances of distinct constructions, namely `Intransitive` and `Unaccusative`. Because this difference is not derived from specific lexical fillers, if the verbs in the examples were to be changed to *dance* and *slip* respectively, the same constructions would be used to describe the sentences.

(1) *Ele correu hoje pela manhã.*
He run.PST.3SG today for morning
'He ran this morning.'

(2) *Ele morreu hoje pela manhã.*
He die.PST.3SG today for morning
'He died this morning.'

In this paper, we discuss insights from negative results obtained in an experiment for identifying schematic clausal constructions and their construction elements in Brazilian Portuguese (pt-br) by using a combination of Multilingual BERT (Devlin et al., 2019) with the computational representations of such constructions in the FrameNet Brasil Constructicon (FN-Br Ccn) (Torrent et al., 2018; da Costa et al., 2018; Almeida and Torrent, 2021). Qualitative analysis of the results indicate that alternatives to the linearization of the constructional properties modeled in resources, number of annotated sentences and a post-processing module

Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023), pages 96 - 109
March 9-12, 2023

should be explored in the future as a means to make use of information in Constructicons for NLP tasks.

In the remainder of this paper, we present, in section 2, how constructions are represented in the FN-Br Ccn. Next, in section 3, we go through the steps needed to convert the FN-Br Ccn representations into a dataset that could be used for proposing the construction identification model in section 4. Sections 5 and 6 describe the experimental setup used to evaluate the model and the results. Discussion of the results is carried out in section 7, with quantitative and qualitative analyses. Section 8 presents final considerations.

2 Construction Representation in the FN-Br Constructicon

The FN-Br Constructicon (Torrent et al., 2014, 2018) is built as part of the FrameNet Brasil language resource, meaning that, similarly to lexical units, constructions in this database can have their meaning import represented in terms of frames. Therefore, the semantics of the `Intransitive` construction licensing (1) can be represented as the `Intentionally_act` frame in Figure (1).

The database structure of FN-Br allows for construction elements (CEs) to be mapped to frame elements (FEs), when relevant. Hence, the SUBJECT and the PREDICATE CEs in the `Intransitive` construction can be respectively mapped to the AGENT and ACT FEs in the `Intentionally_act` frame.

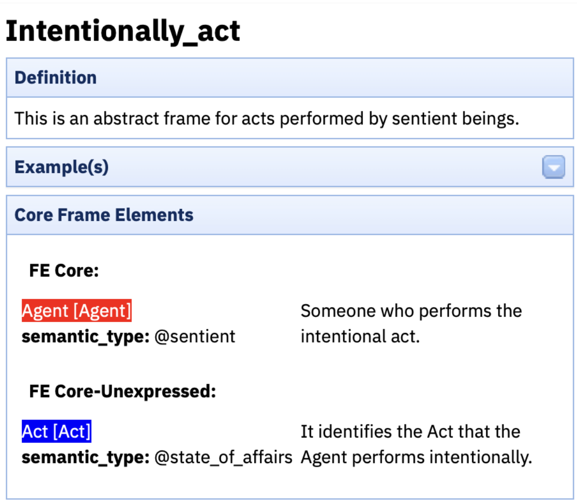

Figure 1: The `Intentionally_act` frame.

Moreover, the FN-Br Ccn allows for other types of information to be represented. First, CEs can be defined in terms of phrasal constructions licensing them. For the `Intransitive`, the SUBJECT CE is a `Determined_noun_phrase`, while the PREDICATE is a *`Non_complement_taking_verb_phrase`*. Furthermore, the information that the verb CE of this last construction has to be filled by a frame that inherits `Intentionally_act` can also be recorded. If instead, this slot was constrained by a child frame of `Undergoing`, then this would be an *`Unaccusative`* construction. Formal properties of the construction can also be represented, such as the fact that the SUBJECT CE usually comes before the PREDICATE, and that the first corresponds to the *nsubj* relation in the Universal Dependencies tag set (de Marneffe et al., 2021), while the latter would correspond to the *root*. All the information associated to the `Intransitive` construction in the FN-Br Ccn, together with the fact that it inherits a general `Subject_predicate` construction are shown in Figure 2.

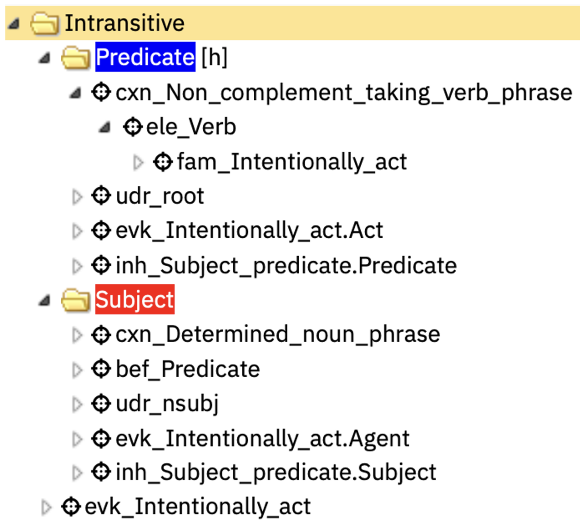

Figure 2: The `Intransitive` construction.

In addition to the two clausal constructions mentioned so far, work by Almeida (2022) has modeled 22 other clausal constructions and 22 phrasal and POS constructions licensing the CEs in them in the FN-Br Ccn. Many of those CEs share the same name (*e.g.* PREDICATE), but are fully separate entities in the database, each belonging to a single construction. For that reason, when applying these structures to an experiment for automatically identifying construction in corpora, the CEs can be treated as the actual labels. A model working with CEs is, arguably, more informative and easier to interpret, despite being more complex. Moreover, the frame information, used to represent the semantic part of a construction, is not lost because the CEs are directly related to the FEs of frames. The full dataset that includes pieces of the FN-Br Ccn and setup used in this work are described next.

3 Dataset

The dataset used in the experiments had to be built step by step because one of our research goals was to assess the impact of Universal Dependencies (UD) and Frame information embedding into a neural model for CE labeling, which is not a traditional NLP task. The corpus consists of 673 sentences annotated for UD, clausal constructions (and their CEs) and frames. Subsections 3.1, 3.2 and 3.3 describe how each type of data was integrated into the same dataset.

3.1 Universal Dependencies Treebank

To evaluate the impacts of UD information when labeling CEs in sentences, the model must be trained on a corpus that has both types of data. Instead of using the annotated sentences from the FN-Br Ccn and including UD annotations, we opted to use an existing, manually-annotated UD treebank and include constructional information. Using a corpus that has been reviewed by specialists reduces the chances of results being affected by poor quality UD annotations. Moreover, a manually-annotated treebank has the advantage of guaranteeing that the model results are be influenced by another system's errors. For those reasons, the UD (Brazilian) Portuguese GSD treebank was chosen [1]. It comprises 12019 sentences and 297045 tokens and was originally annotated using Stanford-style dependencies for multiple languages and later converted into UD (McDonald et al., 2013).

3.2 Constructions

To annotate the constructions for the UD pt-br GSD sentences, the FN-Br WebTool was used, as it already contains the required set of features to work with constructions and visualizing them (Torrent et al., forthcoming). We worked exclusively on the test subset of the UD pt-br GSD treebank, containing 1200 sentences. Before the annotation process was carried out, 24 construction elements from 11 argument structure constructions were selected for annotation. This set was chosen among all of the constructions modeled by Almeida (2022) because they were more likely to occur in the GSD treebank. Moreover, our aim was to identify highly schematic constructions, in opposition to constructions with many fixed slots that could be identified by hybrid or rule-based systems. In total, 673 sentences were annotated. Table 1 shows not only the counts for each construction, but also their schemata (for instance examples, see Appendix A).

It is worth noting that the `Instransitive` and `Ergative` pair discussed in section 1 is not the only in which constructions share a schema. The same happens to the `Indirect_transitive` and the `Oblique_transitive`, but the former is used by dative indirect objects, while the latter is more general. The difference between the `Elapsed_time` construction and the `Presentational_existential`, as their names suggest, is semantic. The former confirms that something happened a certain time ago, while the latter simply introduces a new entity or event to a discourse. Finally, the `Stative_nominal_predicative` and `Attributive_nominal_predicative` constructions assign states or attributes to their SUBJECTS, something closely related to the type of verbal copula present in the sentence. Other constructions are constrained by the presence existential verbs, indicating that the task of labeling CEs deals with lexical, semantic and syntactic constraints simultaneously.

In regards to their elements, the majority of the constructions considered for the experiments have only their SUBJECT and PREDICATE CEs (which are treated as distinct types of subject and predicates), with the execption of `Elapsed_time` and `Presentational_existential`, which have EXISTENTIAL VERBS, NOMINALS and SECONDARY PREDICATES. Because the variety of pt-br in the UD GSD tends to be monitored for verb inflection and SUBJECTS could be nully instantiated, in some sentences, only PREDICATE CEs were annotated. The annotation schema was designed to handle those cases. It is also worth noting that multiple constructions can occur in one single sentence. However, those instances were discarded in next steps, so that the model could be trained to label a single CE (see section 5).

[1] https://github.com/UniversalDependencies/UD_Portuguese-GSD

Construction	Schema	# Sent
`Active_bitransitive`	[NP [V NP [PP]]]	21
`Active_direct_transitive`	[NP [V [NP]]]	337
`Indirect_transitive`	[NP [V [PP]]]	7
`Oblique_transitive`	[NP [V [PP]]]	75
`Intransitive`	[NP [V]]	33
`Ergative`	[NP [V]]	30
`Elapsed_time`	[V_{exi} [NP [VP]]]	2
`Presentational_existential`	[V_{exi} [NP [VP]]]	8
`Locative_predicative`	[NP [V_{cop} [AdvP | PP]]]	17
`Attributive_nominal_predicative`	[NP [V_{cop} [AP | NP]]]	106
`Stative_nominal_predicative`	[NP [V_{cop} [AP | NP]]]	37
Total	**-**	**673**

Table 1: Constructions present in the dataset with their respective schemata and number of annotated examples. The subscripts specify that the slots must be filled by existential verbs or verbal copulas. With the exception of `Elapsed_time` and `Presentational_existential`, all constructions have SUBJECT and PREDICATE construction elements. The CEs on these two are the EXISTENTIAL VERBS, NOMINALS and SECONDARY PREDICATES.

3.3 Frames

The FN-Br Ccn represents constructions in an interconnected graph to express inheritance between them, but also to connect them to other types of entities, including frames, which can be used to explicitly define the semantics of constructions (see Appendix B). Although it would not make sense to feed this frame information to our model because it is part of the prediction objective, these frames serve as anchor nodes to identify relevant clusters in the network. Such clusters can be used to improve the quality of CE classification. The idea of using frame clusters as explicit semantic information was implemented using two algorithms that compute potentially relevant frames for each token in the sentences.

The first algorithm, responsible for frame disambiguation, has been used in previous works (Matos and Salomão, 2014; Costa et al., 2022). It consists of a variation of the spreading activation algorithm executed over the whole network. First, the system identifies and activates the nodes for the words in the sentence, then it iterates over their neighbor nodes spreading "energy". For each word that contains a potential lexical unit, the frame with the highest energy is selected as the evoked frame. The algorithm is highly dependent on FN-Br's coverage, especially of lexical items, because they act as the initial activation points.

The goal of the second algorithm is to identify a set of frames related to a token that could be relevant for label prediction. The procedure depends on a fixed set of frames, containing those related to one of the 11 relevant argument structure constructions in the database. FN-Br is also modified when running this algorithm: it is transformed in a digraph where arcs represent the inheritance, subframe and perspective relations in the original database. For each token, the system finds the minimum paths from its frame to the frames related to constructions in the digraph. In many instances, this path doesn't exist and the token is associated only to its own frame. For the others, the frame is associated to the whole cluster of potentially relevant frames.

4 Model

Figure 4 shows the general architecture of the proposed model. The system was designed and implemented in a way such that some components could be switched or just removed to facilitate testing of the various scenarios.The most important elements in the model are described next.

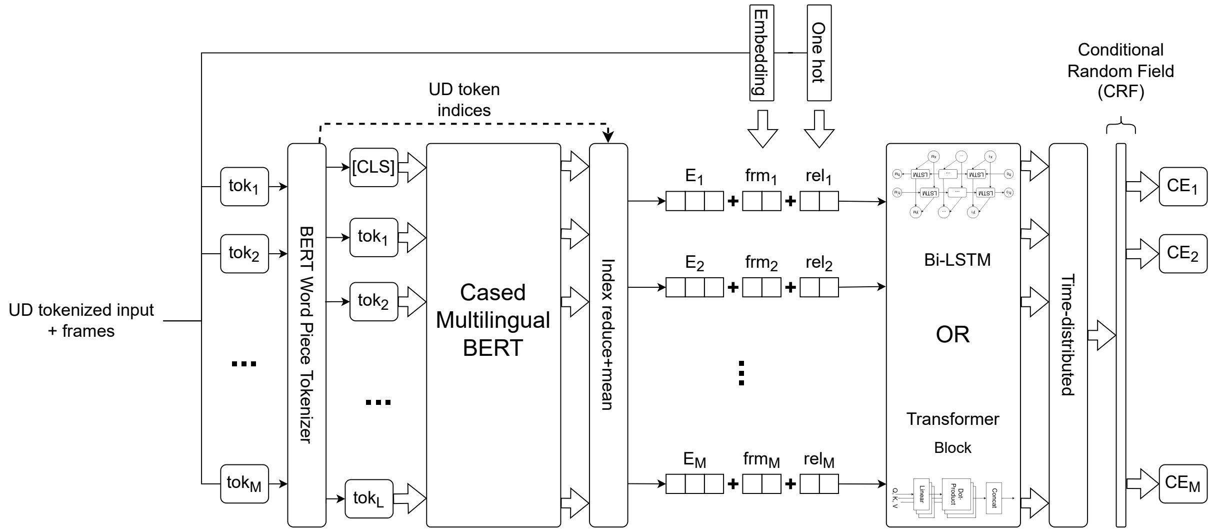

Figure 3: The complete architecture of the proposed system. Because BERT manipulates word piece tokens, the sequence size of the UD annotated sentence (M) is not the same as L. The output from BERT is transformed into a M size sequence later in the pipeline using a mapping of word piece indices to UD tokens. Each vector i in this sequence – referred as E_i in the image – is concatenated with its position frame embedding and UD relation, which is then used by other components to label the construction element in that position.

4.1 Preprocessing

As described in section 3, the dataset built for the experiment already includes tokenized sentences. In this schema, tokens correspond to words, with the exception of some special cases, such as contractions. BERT, however, is trained on sequences created by a word piece tokenizer, *i.e.*, tokens can be full words, but also subwords. During preprocessing, each sentence in the corpus went through BERT's word piece tokenizer and the resulting sequences of subwords were stored. Using those sequences and the treebank tokenized sentences, a mapping between indices was computed for each record, so that, given any subword, its complete token can be retrieved. Both the BERT tokenized sequences and the mappings serve as inputs to the model.

4.2 Encoding UD relations & Frames

Neural networks can process syntactic trees using two main approaches: having a specialized architecture to handle these complex data structures or apply some form of transformation to linearize the trees (Tai et al., 2015; Liu et al., 2017). The former has the advantage of being designed to perform this type of task, albeit being more computationally expensive and more complex to implement. In this work, the trees were linearized using a strategy very similar to the one described by Liu et al. (2017). It works by first associating each token with its one-hot encoded relation to its head. By itself, this is not enough to represent the relation because there is no information about the head. To compensate for this, the tokens are reorganized into a Breadth-first search (BFS) sequence order, which guarantees that the head of a relation will always come before its dependent tokens. The only setback is the lack of limits in the distance between two related tokens. It is important to note that this reordering of the sequence never happens before the sentence is processed by BERT, as that is not compatible with how the language model was trained.

Similarly to the UD relations, the frame clusters associated to each token in the dataset were linearized into sparse binary vectors where each position indicates the presence of a frame. Those sparse vectors of size 1136 (total number of frames) are reduced to 50-dimension vectors by a dense layer before they are used by a LSTM or Transformer Block. This linearization process does not embed any type of information about the relation between frames, but has the advantage of being easily integrated into the model without the need of a special architecture.

4.3 Pre-trained BERT

In all of our experiments, a pre-trained multilingual BERT model (Devlin et al., 2019) was used as the first component, with the goal of obtaining a sequence of vectors from a sequence of sub-words in our corpus. Although word embedding models could have been used for this step, there are advantages in using a language model. First, because of how those models are trained, vector representations of tokens are contextual, *i.e.*, there is no single word vector, but a representation of that word in a specific sentence. The added information is especially useful, considering that our final task is the identification of CEs that may be represented by multiple words in a sentence. In fact, there is evidence that constructional information can be identified and extracted from BERT vectors (Tayyar Madabushi et al., 2020). Second, the fact that a single model was trained in 104 languages makes it easier to evaluate our experiments for other Constructicons, modeled after other languages. Finally, the applicability of this type of model to many different tasks in NLP makes it a good candidate for fine-tuning in our CE labeling experiments.

For all settings presented in section 5, the BERT model was fine-tuned to each downstream task using the multilingual cased parameters as the checkpoint[2]. In this procedure, each BERT sequence output is transformed into one of smaller dimensions before feeding it to the subsequent layers. This transformation was necessary because BERT operates at the subword level, while our CE labels are assigned at the word level. For this step, we simply averaged all of the subword vectors of a single word to obtain a sequence of a smaller size.

4.4 Bi-LSTM

Long short-term memory (LSTM) artificial neural networks are designed to process sequences of data without the caveats of normal recurrent networks, especially the problem of vanishing gradients (Hochreiter and Schmidhuber, 1997). A LSTM unit processes data in sequential timesteps, taking as input the cell and hidden state from a previous timestep, as well as the actual data input and outputting new cell and a new hidden output. In theory, each output is related to a different type of information: the hidden state, when dealing with text data, is the current token output and the cell state is a more general, sentence-level memory that can always be influenced.

In our experimental setup, we used unmodified LSTM cells, containing only the forget and input gates to change the cell state, and the output gate. We also made sure to use a Bidirectional LSTM, since relevant information of a CE can be present before or after the actual CE in the sentence. During development, we have decided to use hidden (and cell) states of 20 dimensions for each direction, because greater values didn't increase performance. The forward and backward hidden states were concatenated, resulting in vectors of size 40 for each position in the sequence. In the final model, the Bi-LSTM layer input are the averaged BERT vectors concatenated with their UD and frame information and in BFS order, according to their dependency tree. This layers transforms the inputs to vectors of lower dimension to be classified by a final layer.

4.5 Transformer Block

The Transformer architecture (Vaswani et al., 2017) was proposed as a "simpler" alternative to popular sequence neural networks, relying only on attention mechanisms, instead of the recurrence observed in an LSTM network, for example. The most important mechanism in a Transformer Block is the Multi-Head Self-Attention, a series of computations that generate multiple weight matrices—generally referred to as attention filters—used to transform parts of the input based on the whole input itself. Each attention filter captures a different aspect of the information and their results are then concatenated. In NLP, this mechanism is usually exemplified as the importance that each word in a sentence has for every single word, where importance can be framed in various ways.

In BERT's architecture, the Transformer is the main unit. Hence, the use of an additional layer in our proposed model can be seen as an extension to make the language model fit the goals of our experiment. The difference between BERT's original layers and the one included in this work is on the hidden dimension size and the type of input sequence. The block still has 12 attention heads, but they manipulate hidden vectors of size 300, instead of the 768 in Multilingual BERT. This reduction was mostly motivated by hardware limitations, but also because the layer is closer to the actual output of the system, which is way smaller in dimension. In regards to the input sequence, this Transformer takes as input a sequence with the same size as the UD token sequence, not the one used by BERT. Each position in this sequence consists of the averaged BERT subword vectors concatenated with the UD relation and frame information, similarly

[2] https://github.com/google-research/bert

to the LSTM.

4.6 Conditional Random Fields

Conditional Random Fields (CRF) are a class of discriminative models that can classify a sample considering its contextual information. In NLP, this type of model has been used extensively for labelling tasks, such as POS tagging and NER (Chiche and Yitagesu, 2022; Li et al., 2020). Similarly to the latter, in our experiments, we decided to use a CRF layer after the final Bi-LSTM (or Transformer) layer because a CE generally spans more than a single token of the input sentence. While a simple dense layer applied to all tokens can independently predict CE labels, the CRF is parameterized to capture the internal logic of labels, which can correlate to construction constraints. For example, in the vast majority of cases in Brazilian Portuguese, a PREDICATE CE cannot be followed by a SUBJECT CE. Moreover, it can attenuate mistakes made by the model in previous layers by using both linguistic information and the labeling probabilities. In the experiments where the CRF was used, the log-likelihood was used as the loss function.

5 Experiments

In order to understand the impacts of UD and frame data in CE labeling, 9 different experimental setups were proposed, 5 variations using LSTM and 4 using Transformers. For each of those options, the effectiveness of the CRF was evaluated, with and without UD and frame data. The LSTM was the only model where the BFS sorting of tokens was tested, hence it has one more variation. This type of ordering was considered only for this architecture because, in theory, the way information vanishes in the cell states is influenced by the order of the elements in the input sequence. The Transformer can handle this problem by simply adjusting attention weights.

One of the challenges of working with the dataset described in section 3 was the number of samples annotated for each construction. For `Elapsed_time`, for example, only two examples were found in the 673 sentences. This variation in construction frequency is expected and leads to the fact that a much larger dataset would be needed to find a reasonable amount of examples for that construction. We have decided to consider only the two most frequent constructions in the UD Portuguese GSD treebank, namely `Direct_transitive` and `Attributive_nominal_predicative`, as the models could not perform consistently for the ones with less samples. This resulted in a dataset with 443 sentences and 4 CE labels that was split into train and test sets in a 8:2 ratio. Considering the relative effectiveness of BERT's fine-tuning and that constructional information can be extracted from it, this dataset can still be used to predict CE labels (Devlin et al., 2019; Sun et al., 2019; Tayyar Madabushi et al., 2020).

In all variations, the networks were implemented to predict a single CE label (or none) for each position corresponding to a token in the sequence, despite the fact that it is possible for more than one label to be true. This was done because, in our first implementation tests, we verified that only one instance of the dataset had a token with two labels. Moreover, convergence was slow during training, even after adjusting parameters, without any performance gains. For that reason, we used a softmax activation function and cross-entropy as the loss function. When the final layer was a CRF, loss is computed using the log-likelihood.

For training, we used an Adam optimizer with learning rate set to $3e-5$. Due to GPU memory limitations, batch size was set to 16 samples and the maximum number of epochs to 20, which was not a problem because of the reduced size of the training dataset. To prevent over-fitting, the loss over the validation set was monitored and after 3 epochs without any improvements, training would be stopped, resulting in less than 20 epochs per training run. Every model variation was trained 10 times so that their performance and generalization could be better analyzed. The BERT model's weights were adjusted, *i.e.* fine-tuned, in each of those runs and, in order to prevent tests from influencing one another, memory was cleaned up between executions.

Table 2 summarizes the number of parameters in each model as the difference to each of the two base model types and their average number of training epochs.

6 Results

After the execution of all of the training algorithms, model results were computed and compiled into Table 3. The main metric used for evaluation was a macro-F1 calculated by treating each label as a

Model	Δ|θ|	epochs
LSTM		
Base	0 (~180M)	10.5
CRF	+35	11.7
Frm, UDrel	+70,450	10.0
Frm, UDrel+order	+70,450	10.3
Frm, UDrel, CRF	+70,485	11.5
Transformer		
Base	0 (~206M)	10.4
CRF	+35	12.0
Frm, UDrel	+6,725,525	10.0
Frm, UDrel, CRF	+6,725,560	11.6

Table 2: Model size in number of parameters and number of epochs used on average for training.

Model	F1	
	μ	best
LSTM		
Base	.694 (.050)	**.767**
CRF	**.700 (.015)**	.720
Frm, UDrel	.647 (.081)	.709
Frm, UDrel+order	.603 (.111)	.763
Frm, UDrel, CRF	.675 (.072)	.748
Transformer		
Base	.643 (.044)	.703
CRF	.643 (.044)	.720
Frm, UDrel	.618 (.033)	.653
Frm, UDrel, CRF	.638 (.054)	**.767**

Table 3: Average and best macro-F1 scores for each model, based on the results of 10 separate training executions. Standard deviations are shown in parentheses. The best overall results for all experiments are highlighted.

binary class, computing their F1s and then averaging. The label used to indicate the absence of a CE is ignored in this calculation. The main advantage of using a macro-F1 over the micro-F1 or accuracy lies on the fact that the absence of a CE can be treated asymmetrically. This is relevant for our analysis because it can focus on the predictions where the model assigned a label in order to obtain insights.

To better understand the variations between different training iterations, the average (with standard deviation) and the best F1 scores for each configuration were observed. In terms of averages, the LSTM model with a CRF, but without frames or UDs had the best performance. This configuration also had the smallest standard deviation, indicating that training is somewhat consistent. In terms of best results, the base LSTM model without CRF and the complete Transformer model have the highest F1, with a score of .767. Of the two, the LSTM is a considerable smaller model, as shown in Table 2. Taking into consideration that the averages of the models are not that different, for a LSTM-based model, the inclusion of only a CRF seems to yield the best results. For a Transformer-based network, the extra semantic and syntactic information, along with the CRF, contributes to better results.

The worst configuration, on average, was the LSTM model where tokens were reordered using the dependency tree BFS results. In constrast, every model achieved better average F1 scores when a CRF layer was added.

7 Discussion

As previously stated, the LSTM models performed better, especially when no additional frame or UD data was embedded into the inputs. However, when using Transformers, the same type of data can increase performance. One possible explanation is that the latter has a considerably larger number of parameters, making it easier to integrate the additional information, but, at the same time, being more complex to train and, thus, having worse performance than LSTMs. Also, the difference in the results is likely affected by the small size of the training dataset, it is possible that the quality of the predictions could improve if more samples were processed by the networks.

The LSTM where the order of the tokens was changed also provides good insight on how this model used the information to make predictions and why it has the lowest average F1 score. When analyzing the average F1 scores for each construction element, it was noted that final F1 was mostly influenced by the SUBJECT CE of the `Attributive_nominal_predicative`. In the model using a CRF, the F1 for the subject was .628, for the one with the BFS ordering, it was .388. This is a strong evidence that when optimizing with few samples, for a CE that has a relatively strict position in a sentence, positional information is relevant. It also shows that the model was not able to compensate for the absence of this kind of information using only UDs and a different ordering. This type of problem can be potentially avoided by using a network architecture designed to handle graphs or trees.

We have also decided to carry out a qualitative analysis of the predictions made by the best Transformer model. All of the predictions made over the test set were transformed back to CE spans, which where then aligned to the original sentences and paired with the original human annotations. Making this side-by-side comparison, notes were taken for each record. During this process, we observed that some types of errors were way more frequent than others. For instance, 26% of the sentences had only the head words of the constructions elements labeled, while 13% had a problem of discontinuity in the CE span. These numbers agree with the F1 results displayed on Table 3 that show an improve in performance when a CRF layer is added. Because this type of layer models the relation between the classification labels, it is able to correct some of the mistakes in continuity and length of the CE spans made by the previous layers.

More importantly, these errors seem to originate from an overgeneralization made by the model over the POS of words. Despite the fact that POS tags are not part of the input to the model, this information is arguably embedded into BERT. More evidence of that is found on examples where the model labeled some word with the incorrect CE. Although rare, when it happens, the CE predicted for that word is of the same POS of a head of that CE. For example, many verbs are labeled as the predicate of a `Direct_transitive` construction, even when they are part of other type of construction. The same happens for adjectives and the `Attributive_nominal_predicative`. This happened in 8.7% of the analyzed sentences.

The problem of overgeneralization also occurs with the *conj* relation in this model. Interestingly, this seems to be the only UD relation that clearly influenced the predictions of the test set. In 8,7% of the sentences, the model labeled the tokens of a conjunct despite the fact that they are not related to a subject or predicate. In a deterministic approach, this type of error can be easily verified using the dependency trees, as the CE span nodes would not be connected.

8 Final considerations

The experiment reported on in this paper aimed at testing whether UD and frame information extracted from a Constructicon could positively influence the performance of Transformer-based models for schematic construction identification in sentences. However, the most effective models in our tests are still the smaller LSTMs, without any extra information. Furthermore, we have identified that the trained models were overgeneralizing certain aspects of the data, causing performance to degrade.

One of the limitations of our experiments is in the architecture of the network itself when these various types of information are used. For both LSTM and Transformers, there are gaps that could be filled if they were implemented to fully handle tree and graph structures instead of their linearized, thus, simplified, versions. Other changes in the network structure and training procedure are needed to prevent the overgeneralization discussed in section 7.

Another course of action to better understand how neural networks can be used to classify constructions effectively is to expand the dataset, both in number of samples, but also in representation of different clausal structures. For example, if the models were to be trained to also label the CEs of the `Unaccusative` construction, they would have to learn the semantic boundaries that differentiate an unaccusative from a transitive verb.

Finally, in spite of their limitations, it is clear that performance could be improved by using post-processing algorithms that could either find anomalous outputs (*e.g. the incorrect conjucts* and even expand the spans of the CEs based on the dependency relations. This type of procedure is adequate because the models are already able to identify most CE heads.

References

Vânia Almeida and Tiago Torrent. 2021. Construções de estrutura argumental com argumento preposicionado: uma modelagem linguístico-computacional na FrameNet Brasil. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 353–362, Porto Alegre, RS, Brasil. SBC.

Vânia Gomes de Almeida. 2022. *Modelagem e Identificação Automática de Construções de Estrutura Argumental: uma proposta para o Constructicon da FrameNet Brasil*. Ph.D. thesis, Graduate Program in Linguistics, Federal University of Juiz de Fora, Brazil.

John Edward Bryant. 2008. *Best-Fit Constructional Analysis*. Ph.D. thesis, EECS Department, University of California, Berkeley.

Alebachew Chiche and Betselot Yitagesu. 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1):1–25.

Alexandre Diniz da Costa, Mateus Coutinho Marim, Ely Matos, and Tiago Timponi Torrent. 2022. Domain adaptation in neural machine translation using a qualia-enriched FrameNet. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1–12, Marseille, France. European Language Resources Association.

Alexandre Diniz da Costa, Vânia Almeida, Ludmila Lage, Gustavo Barbosa, Natália Marção, Vanessa Paiva, Ely da Silva Matos, and Tiago Torrent. 2018. Representação computacional das construções de sujeito-predicado do português do brasil. *Revista LinguíStica*, 14(1):149–178.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

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.

Charles J. Fillmore. 1968. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*, pages 0–88. Holt, Rinehart and Winston, New York.

Charles J. Fillmore. 2008. Border conflicts: FrameNet meets Construction Grammar. In *Proceedings of the XIII EURALEX International Congress*, pages 49–68, Barcelona. Universitat Pompeu Fabra, Universitat Pompeu Fabra.

Adele E. Goldberg. 2013. Constructionist approaches. In Thomas Hoffman and Graeme Trousadale, editors, *The Oxford Handbook of Construction Grammar*. Oxford University Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. 2017. Structural embedding of syntactic trees for machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 815–824, Copenhagen, Denmark. Association for Computational Linguistics.

Benjamin Lyngfelt, Lars Borin, Markus Forsberg, Julia Prentice, Rudolf Rydstedt, Emma Sköldberg, and Sofia Tingsell. 2012. Adding a constructicon to the Swedish resource network of Språkbanken. In *Proceedings of KONVENS 2012*, pages 452–461. ÖGAI. LexSem 2012 workshop.

Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent. 2018. *Constructicography: Constructicon development across languages*. John Benjamins, Amsterdam.

Ely Matos, Tiago Torrent, Vânia Almeida, Adrieli Laviola, Ludmila Lage, Natália Marção, and Tatiane Tavares. 2017. Constructional Analysis Using Constrained Spreading Activation in a FrameNet-Based Structured Connectionist Model. In *The AAAI 2017 Spring Symposium on Computational Construction Grammar and Natural Language Understanding Technical Report SS-17-02*, pages 222–229, Palo Alto, CA. AAAI Press.

Ely Edison da Silva Matos and Maria Margarida Martins Salomão. 2014. Ludi: um framework para desambiguação lexical com base no enriquecimento da semântica de frames. *Revista Linguística*, 12(1).

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.

Kyoko Hirose Ohara. 2014. Relating frames and constructions in Japanese FrameNet. In *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014, pages 2474–2477. European Language Resources Association (ELRA). Copyright: Copyright 2017 Elsevier B.V., All rights reserved.; 9th International Conference on Language Resources and Evaluation, LREC 2014 ; Conference date: 26-05-2014 Through 31-05-2014.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tiago Timponi Torrent, Ludmila Meireles Lage, Thais Fernandes Sampaio, Tatiane da Silva Tavares, and Ely Edison da Silva Matos. 2014. Revisiting border conflicts between FrameNet and Construction Grammar: Annotation policies for the Brazilian Portuguese Constructicon. *Constructions and Frames*, 6(1):34–51.

Tiago Timponi Torrent, Ely Edison Matos, Alexandre Diniz Costa, Maucha Andrade Gamonal, Simone Peron-Corrêa, and Vanessa Maria Ramos Lopes Paiva. forthcoming. A Flexible Tool for a Qualia-Enriched FrameNet: the FrameNet Brasil WebTool. *Language Resources and Evaluation*.

Tiago Timponi Torrent, Ely Edison Matos, Ludmila Meireles Lage, Adrieli Laviola, Tatiane da Silva Tavares, Vânia Gomes de Almeida, and Natália Sathler Sigiliano. 2018. Towards continuity between the lexicon and the constructicon in FrameNet Brasil. In Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent, editors, *Constructicography: Constructicon development across languages*, pages 107–140. John Benjamins, Amsterdam.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alexander Ziem and Hans Boas. 2017. Towards a Constructicon for German. In *The AAAI 2017 Spring Symposium on Computational Construction Grammar and Natural Language Understanding Technical Report SS-17-02*, pages 274–277, Palo Alto, CA. AAAI Press.

A Argument structure constructions in the FN-Br Ccn

A.1 Active_bitransitive

This construction expresses predicates with three central participants, *i.e.* a trivalent event.

(3) *O jornal atribui o*
The news attribute.PRS.3SG the
abandono ao custo da
abandonment to_the cost of_the
ferrovia.
railroad

'The news attributes the abandonment to the cost of the railroad.'

(4) *O ministro transferiu a*
The minister transfer.PST.3SG the
sede da colônia para o
head_office of_the colony to the
Rio de Janeiro.
Rio de Janeiro

'The minister transferred the seat of the colony to Rio de Janeiro.'

A.2 Active_direct_transitive

This construction is licensed by predicates that require at least two participants, one agent and the other is patient-like.

(5) *A agência federal determinou*
The agency federal determine.PST.3SG
o início imediato dos trabalhos.
the start immediate of_the works

'The federal agency determined the immediate start of the works.'

(6) *Eu misturo o tempero e está*
I mix.PRS.1SG the seasoning and it's
pronto!
ready

'I mix the seasoning and it's ready!'

A.3 Indirect_transitive

This construction is very similar to the `Oblique_transitive` because both have the predicated object introduced by a preposition. The main difference is that the indirect object in this construction must be a dative object, *i.e.* it needs to play a beneficiary or recipient role.

(7) *O diretor respondeu aos*
The director reply.PST.3SG to_the
jornalistas.
journalists

'The director answered the journalists.'

(8) *Assim_que a carta chegou,*
As_soon_as the letter arrive.PST.3SG,
contaram para ele.
tell.PST.3PL to he

'As soon as the letter arrived, they told him.'

A.4 Oblique_transitive

The `Oblique_transitive` construction is the one used by certain verbs in Portuguese, in which the oblique/indirect object is introduced by a prepositional phrase. These complements are not optional and, semantically speaking, the event has two central participants.

(9) *A cidade precisa de uma reflexão mais profunda.*
The city need.PRS.3SG of a reflection more deep

'The city needs a deeper reflection [on the matter].'

(10) *A família procurou por cirurgias corretivas*
The family look.PST.3SG for surgeries corrective

'The family sought corrective surgeries.'

A.5 Intransitive

Construction with an agent-like subject and an unergative verb.

(11) *João Paulo concordou com a fala.*
João Paulo agree.PST.3SG with the statement

'João Paulo agreed with the statement.'

(12) *Eu ensinei com entusiasmo.*
I teach.PST.3SG with enthusiasm

'I taught with enthusiasm.'

A.6 Ergative

Construction with a non agent-like subject and an unaccusative verb.

(13) *O jogo começou em ritmo alucinante.*
The game start.PST.3SG on pace crazy

'The game started at a breakneck pace.'

(14) *A produção industrial aumentou 1,7% ante abril.*
The production industrial rise.PST.3SG 1.7% from april

'Industrial production rose 1.7% from April onwards.'

A.7 Elapsed_time

In this construction, the idea that an event occurred some time ago is expressed. In the following examples, the verb 'haver' was translated as 'have', but has the meaning of an exlcusively existential verb.

(15) *Eles moram naquela cidade há vinte anos.*
They live.PRS.3PL in_that city have.PRS.3SG twenty years

'They have lived in that city for twenty years.'

(16) *O crime aconteceu há três dias.*
The crime happen.PST.3SG have.PRS.3SG three days

'The crime happened three days ago.'

A.8 Presentational_existential

This type of construction is used to add new, entity-central, information to a discourse, *i.e.*, there's no subject being referred to, only existential predication of a nominal. Optionally, this nominal can be followed by a secondary predicate.

(17) *Existem 30 negócios na categoria.*
exist.PRS.3PL 30 businesses in_the category

'There are 30 businesses in the category'

(18) *Tem umas pessoas esperando você lá fora.*
have.PRS.3SG some people wait.PRS.PROG.3SG you there outside

'There are people waiting for you outside'

A.9 Locative_predicative

This type of construction is used to express where the SUBJECT is located.

(19) *E eles estão na cadeia, naquele inferno.*
And they be.PRS.3PL in jail, in_that hell

'And they're in jail, in that hell'

(20) *Seu pai estava em serviço na Coréia.*
his father be.PST.3SG in service in_the Korea

'His father was on duty in Korea.'

A.10 Attributive_nominal_predicative

This type of construction consists of a predicational clause where a stable object or property is being predicated and is quite similar to the `Stative_nominal_predicative`, with the only difference being the stable *vs* temporary construal.

(21) *O apoio dos fãs também será essencial.*
The support of_the fans also be.FUT.3SG essential

'The support of the fans will also be essential.'

(22) *Reichenbach é um município na Alemanha.*
Reichenbach be.PRS.3SG a municipality in_the Germany

'Reichenbach is a municipality in Germany.'

A.11 Stative_nominal_predicative

The type of construction in which a temporary state concept is predicated. In pt-br, the copula 'estar' is not exclusively but usually used for a stative construal of the SUBJECT. Being sad or hungry are very prototypical temporary states, but it is possible to have attribute-like states construed as temporary.

(23) *Os gravetos estavam todos molhados.*
The sticks be.PST.3PL all wet

'The sticks were all wet.'

(24) *Ele fica desconfiado.*
He stay.PRS.3SG suspicious

'He gets suspicious.'

B The 11 argument structure as a subgraph of the FN-Br Ccn

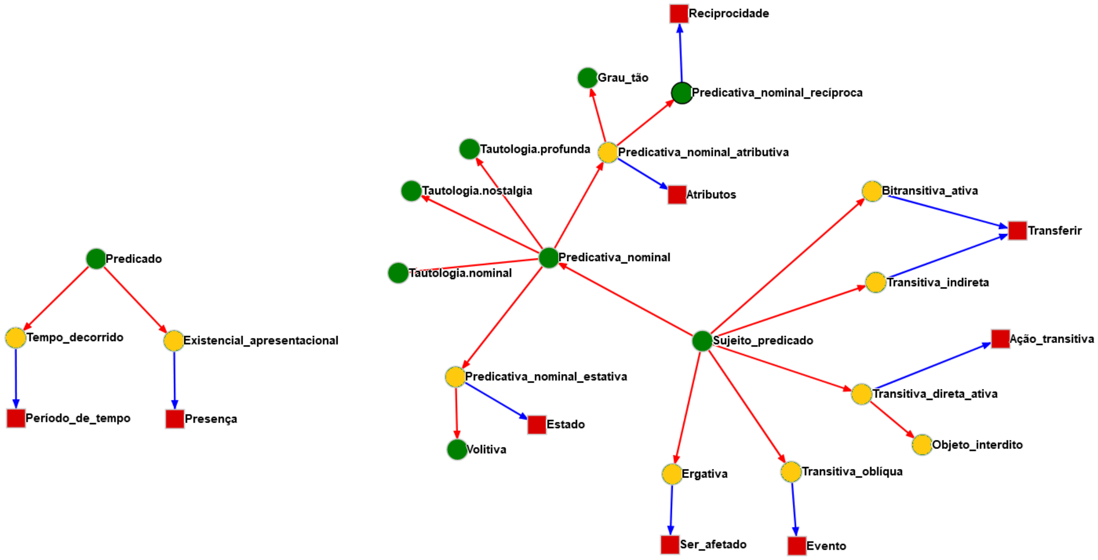

Figure 4: A subgraph of the FN-Br Constructicon containing all of the 11 argument structure selected for this paper. Their nodes are indicated in yellow, while other related constructions are green. Squares represent connections to frames, as discussed in the manuscript. Arrows in red are used for construction inheritance relations and the blue ones for frame evokation.