Generalization through Lexical Abstraction in Transformer Models: The Case of Functional Words

Anonymous ACL submission

Abstract

Pronouns, adverbs and other functional words (such as *they, her, somewhere, there*) are often used in language to replace concrete nouns or phrases, when their properties – such as gender, grammatical number – provide sufficient information for the given context. Do pretrained transformer models encode such functional words in a manner that allows them to be used like humans do? Can language models recognize the syntactic and semantic parallelism of sentences such as "The researchers wrote the paper" and "They wrote it", which relies on such lexical abstraction?

We map these linguistic questions into the embedding space of a pretrained transformer model, and compare representations of nouns, with the representations of the pronouns and adverbs that can replace these nouns, in isolation and in parallel lexicalized and functional sentences. We then probe for shared syntactic and semantic structure in the embeddings of parallel lexicalized and functional sentences.

We find that functional words are located centrally compared to nouns, but are also distinct, which is congruent with their behaviour as place-holders in a wide variety of contexts. The analysis of the embeddings of parallel sentences shows that they do encode the shared syntactic-semantic structure. Moreover, this information is encoded in a similar manner in the representations of functional and lexicalized sentences, thus providing supporting evidence that large language models do encode some form of lexical abstraction.

1 Introduction

Large language models (LLMs) are very successful, and much of their success stems from their ability to induce word or token representations that encode the extremely complex language data, with many generative factors (Bengio et al., 2013). It is an ongoing quest to understand these representations, the kind of linguistic information they encode and the way a system is able to successfully manipulate them to solve a wide variety of tasks. It is difficult to attribute their high performance on numerous linguistic and NLP tasks to the LLMs' understanding of language and its structure (Waldis et al., 2024). One of the criteria for judging the degree of language understanding in LLMs is their capacity to “generalize" well. This question is often approached from a technical, rather than a linguistic, perspective. Generalization is considered a crucial property of a learned model, as it ensures trust in its deployment outside of its training environment – whether this application involves a slightly different task, out-of-distribution data, a different language, or some other level of distinction between the application domain and the one it was trained on (Hupkes et al., 2023).

We focus here on linguistic generalizations and abstractions. For example, speakers can easily strip down a sentence to a basic syntactic-semantic structure, such as *Who did what to whom* or *She put that there* or *She does that sometimes.* The use of pronouns or adverbs to reduce a sentence to a “skeleton” does not rely on using out-of-vocabulary items, as pronouns and adverbials, such as *somewhere/sometime*, are some of the most frequent words in a corpus, and appear in many shared contexts, as their frequent use in coreferring expressions attests. In semantics, pronominal forms are usually treated as variables, placeholders for more structured lexical elements within a sentence and thus highly abstract entities (Büring, 2019).

Is this particular property of functional words – as abstract place-holders for nouns and prepositional phrases – captured in LLMs? We map this question into the embedding space of a pretrained transformer model. Embedding spaces are built on the assumption that similarity and relatedness between words is equated with closeness in the embedding space. As abstract place-holders, functional words appear in shared contexts with a wide variety of nouns and prepositional phrases, thus they are similar to a certain degree. We expect them to be somewhere in the center of the embedding space, so they can be close to a wide variety of nouns and prepositional phrases. In the sentence embedding space, we test whether parallel sentences – such as "The researchers wrote the paper" and "They wrote it" – are encoded with embeddings that are close in space as they share syntactic (the sentence structure and phrase types) and semantic properties (semantic roles). If LLMs capture linguistic abstraction, this shared information – syntactic structure and semantic roles – should be encoded in a similar manner regardless of whether the phrases contain nouns or functional words.

We test these expectations uia purposefully generated dataset on the causative verb alternation, with structure and lexical variation at multiple levels. We extract data in several formats, to explore words in isolation and in context, and the functional and lexicalized versions of sentences. We observe that functional words are rather central in the embedding space compared to nouns, but also isolated, which matches the expectations arising from their behaviour as place-holders in a wide variety of contexts. Analyses of sentence embeddings show that while the functional and lexicalized versions inhabit different areas in the embedding space, their shared syntactic-semantic structure can be detected, and is encoded in a similar manner.

2 Data

Verb alternations require observing at least two related sentences. They show that the same verb can appear in different sentential contexts, with systematic syntactic-semantic mappings of their arguments across the sentences, like a system of equations that all share the same variable bindings.

The dataset is generated from a set of verbs belonging to the change-of-state (COS) and object-drop (OD) classes (Levin, 1993). These classes provide an argument structure minimal pair: they share the same syntactic structure - transitive/intransitive alternation - but differ in their argument structure. The object of the transitive verbs belonging to the COS class bears the same semantic role (Patient) as the subject of the intransitive verb (*The artist opens this door/This door opens*). The transitive form of the verb has a causative meaning. In contrast, for OD verbs the subject bears the same semantic role (Agent) in both the transitive and intransitive forms and the verb does not have a causative meaning (*The artist paints this door/The artist paints*) (Levin, 1993; Merlo and Stevenson, 2001).

We divide words into lexical and functional. Lexical elements, or content words, are an open class of words with a meaningful content, corresponding to concepts or entities and events in the world. The role of the closed class of function words, instead, is to express grammatical functions. We focus specifically on pronouns, and a subset of adverbs, those that can express temporal and spatial concepts. These function words can be used as general placeholders for nouns and prepositional phrases: for instance, *The researchers wrote the article last week* can also be expressed more abstractly as *They wrote it then.*

2.1 Data templates

The dataset comprises instances that follow the Blackbird Language Matrices framework (Merlo, 2023). Each instance is a multiple-choice puzzle and it consists of (i) a rule-generated *context* sequence of sentences that illustrate the encoded phenomenon. The rules are of two types: rules that described the linguistic property under study (verb alternation) and rules that are not related to it (e.g. presence or absence of a prepositional phrase). One sentence that would make the sequence complete is missing, and must be chosen from (ii) an answer set of minimally differing contrastive sentences – one correct, and each of the others violating a sub-rule.

Context set The syntax-semantics features of the verb alternation, and their combination rules, lead to the construction of the context set. Specifically, (i) the presence of one or two arguments and their attributes (agents, Ag; patients, Pat) ; (ii) the active (Akt) or passive (Pass) voice of the verb. The phenomenon-external factors include an alternation between a NP introduced (i) by any preposition (e.g., *in an instant*, henceforth p-NP) and (ii) by the preposition by (e.g., *by chance*, by-NP), but not agentive (e.g., *by the artist*, by-Ag/by-Pat), which remains a confounding variable. The OD context minimally differs from the COS in the last sentence of the context: the subject of the intransitive is an Agent, and not the Patient.

Answer Set All answers have the same structure: (NP V by-NP) consisting of a verb, two nominal constituents (giving rise to a structure of the type NP V NP) and a preposition (by, or the lack of the preposition) between the verb and the second NP. The candidate answers comprise the correct intransitive form of the alternation followed by a by-NP which satisfy the rules of the BLM, and the contrastive incorrect answers obtained by corrupting some properties of the rules (wrong argument, wrong voice of the verb, lack of preposition, wrong nominal constituent of the PP).[1]

The answer set does not change across verb classes, only the label of the correct answer: the correct answer for COS is an error for OD, and viceversa. The BLM-template (context and answers) for COS and OD are presented in Figure 1.

COS context				
1	Ag	Akt	Pat	p-NP
2	Ag	Akt	Pat	by-NP
3	Pat	Pass	by-Ag	p-NP
4	Pat	Pass	by-Ag	by-NP
5	Pat	Pass		p-NP
6	Pat	Pass		by-NP
7	Pat	Akt		p-NP
?	???			

COS answers				
1	Pat	Akt	by-NP	Correct
2	Ag	Akt	by-NP	SSM-Int
3	Pat	Pass	by-Ag	Pass
4	Ag	Pass	by-Pat	SSM-Pass
5	Pat	Akt	Ag	Trans
6	Ag	Akt	Pat	SSM-Trans
7	Pat	Akt	by-Ag	WrBy
8	Ag	Akt	by-Pat	SSM-WrBy

OD context				
1	Ag	Akt	Pat	p-NP
2	Ag	Akt	Pat	by-NP
3	Pat	Pass	by-Ag	p-NP
4	Pat	Pass	by-Ag	by-NP
5	Pat	Pass		p-NP
6	Pat	Pass		by-NP
7	Ag	Akt		p-NP
?	???			

OD answers				
1	Pat	Akt	by-NP	SSM-Int
2	Ag	Akt	by-NP	Correct
3	Pat	Pass	by-Ag	SSM-Pass
4	Ag	Pass	by-Pat	Pass
5	Pat	Akt	Ag	SSM-Trans
6	Ag	Akt	Pat	Trans
7	Pat	Akt	by-Ag	SSM-WrBy
8	Ag	Akt	by-Pat	WrBy

Figure 1: BLM COS and OD contexts and answers.

2.2 Levels of lexical abstraction

To explore generalisation through abstraction, we produce two main variants of the data – a lexicalized one (labelled *Lex*), and a functional one, where functional words replace all content words except the main verb (labelled *Fun*). The lexicalised variant comes in different types (type I, II, III), with varying amounts of lexicalisation, for comparison with the small size inventory of the functional words. The groups are exemplified in Figure 2, together with the generation process presented in the next paragraph. Figure 8 and Figure 9 in the appendix provide examples for type I data for both verb classes.

2.3 Main Dataset

The main dataset is built based on thirty (manually chosen) verbs from each of the two classes discussed in Levin (1993). See Table 2 in the appendix for the full list.

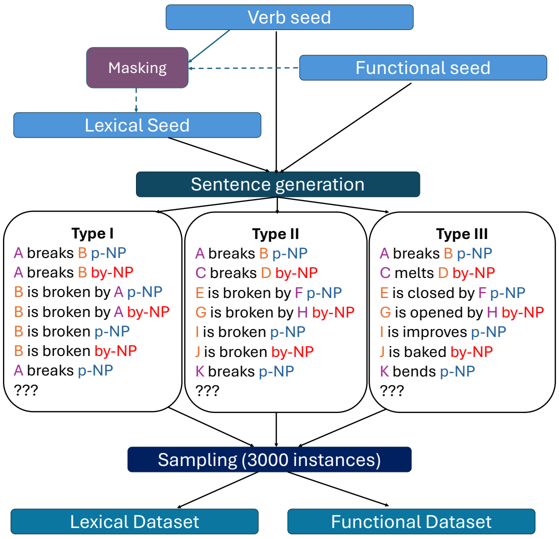

Figure 2: Process of generation of the three levels of lexical variation (type I, II, III), exemplified for COS data. Type I data contains instances with lexically consistent material, with minimal change across the context and the answer set. In type II the verb remains the same while one constituent varies across the context and the answer set. Type III data displays maximal lexical variation in both the context sentences and the answer set.

The functional lexicon has been manually selected by the authors to maintain the syntactic and semantic acceptability of the sentences[2]. The lexical alternatives were provided by a masked language model (*bert-base-uncased*, (Devlin et al., 2018)). The models received sentences containing only the masked constituent, the verb and the functional elements. For example, to retrieve the arguments for the verb *break*, two masked templates are used: the patient is masked and the agent is in pronominal form (e.g. *she broke (the/a/some/...)* <MASK>), and the subject of the transitive is masked and the patient is a pronoun *(e.g. (the/a/some/...)* <MASK> *broke it*. Both the lexical seed and the functional seed contain five semantically plausible instances for each constituent class (Ag, Pat, p-NP and by-NP). We ensured a balanced distribution of tense and number across verbal inflections.

For our experiment, we sampled 3000 instances (out of 38400 combinations of arguments and verbs) for each type, semi-automatically crafted and manually evaluated for plausibility and grammaticality.

[1] Error types: wrong semantic role on the first constituent is a syntax-semantic mapping error (SSM), wrong last constituent introduced by the preposition *by* WrBy*, and the other errors are labelled according to the type of resulting structure – intransitive,* Intr*; transitive,* Trans*; passive,* Pass.

[2] Following the discussion in Haspelmath (1997), we add elements like *somebody* as pronominal elements.

2.4 Dataset variations

Starting from the main datasets described above, we build several variations that will be used in the different experiments.

Words From each sentence in the type I subset of the BLM dataset, we extract the functional words and their corresponding nouns and prepositional phrases. There are 17 functional words and phrases: *he, her, him, it, she, somebody, someone, that, that one, them, these, these ones, they, this, this one, those, those ones* and 204 noun phrases.

Sentences We compile parallel versions of the sentences in their lexicalized and functional word forms from the FUN and LEX subsets of the type I BLM dataset. Each sentence has associated its syntactic pattern (the syntactic version of the syntactic-semantic template shown in Figure 1, e.g. *Pron Vpass PP PP*). From these, we sample 4000 sentences, split 80:20 between training and testing, and use 10% of the training data for validation.

BLM data Of the thirty verbs, all instances for three of the verbs (3x100) are selected for testing. Of the instances of the other 27 verbs, 2000 are randomly sampled for training. Ten percent of the training data is dynamically selected for validation. The same 27:3 verb split is used for all Fun/Lex and type I/ type II/type III variations. All variations have 2000 instances for training, 300 for testing. We also produced a variation where the COS and OD subtasks are merged. The data is split in a similar manner for training and testing.

3 Analyses and experiments

We aim to determine whether language models encode the lexical abstraction property of pronouns and adverbs relative to nouns and noun phrases. We map this question into an analysis of the embedding space of the words and sentences, and proceed in several steps. We investigate the relative positions of lexical and functional word embeddings, when presented in similar sentential contexts. We study the relative positions of the representations of two variations of sentences – with nouns, or with functional words (Section 3.1). We analyse the representation of functional and lexicalized sentences for detecting the shared syntactic structure (Section 3.2). We deploy the BLM linguistic puzzles, whose solution relies on detecting shared structure at the level of input sequence and within each sentence (Section 3.3).

We obtain word and sentence representations (as averaged token embeddings) from an Electra pretrained model (Clark et al., 2020)[3]. We choose Electra because it has been shown to perform better than models from the BERT family on the Holmes benchmark[4], and to also encode information about syntactic and argument structure better (Yi et al., 2022; Nastase and Merlo, 2024).

As a first step of analysis, we analyse 2D UMAP (McInnes et al., 2018) and t-SNE projections of the same data (Hinton and Roweis, 2002). t-SNE is designed to project high-dimensional data into a lower dimensional space while preserving neighbourhood information, while UMAP preserves the global structure of the data. Considering that the embedding space was built based on the notion of similarity and similarity metrics, these two types of visualization provide complementary information about the relative position of the words in our data in the embedding space.

3.1 Contextual word embeddings

We use the parallel versions of the sentences – with content words or functional words – to build contextualized word embeddings, and verify whether the added constraints of belonging in the same sentential contexts brings the word embeddings closer together. Each point in the plots in Figure 3 corresponds to the contextual embedding of a functional word or noun in each of the input sentences.[5]

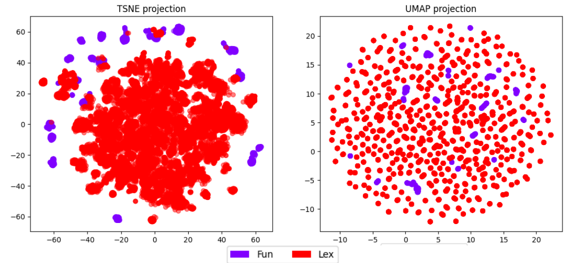

Figure 3: t-SNE and UMAP projections of the embeddings of functional words and nouns obtained from parallel contexts. Each point is a contextual embedding.

The UMAP plot shows the embeddings of the functional words located centrally, which is congruent with their behaviour as place-holders for a wide variety of nouns. The t-SNE plot shows

[3] *google/electra-base-discriminator*

[4] The HOLMES benchmark leaderboard: `https://holmes-leaderboard.streamlit.app/`. At the time of writing, the ranks were: Electra - 16, DeBERTa - 21, BERT - 41, RoBERTa - 45.

[5] Word embeddings obtained for the words in isolation show the same profile (see Appendix B).

them separate from the other nouns, indicating that they also have specific characteristic different from nouns. These observations match the expectations of matching the way these words behave to their relative positions in the embedding space.

3.2 Shared structure in sentence embeddings

The analysis of the relative positions of functional words and nouns in the embedding space supports the hypothesis that the language model encodes functional words in a manner that matches their behaviour as place-holders for nouns. We deepen the exploration by checking whether the embeddings of parallel versions of sentences – with functional words or nouns and prepositional phrases – encode their shared syntactic-semantic information.

Figure 4 shows the t-SNE and UMAP projections of the representations of the two variations of each sentence. The functional and lexicalized version of the sentences occupy different regions of the embedding space, seeming to provide a negative answer to our question.

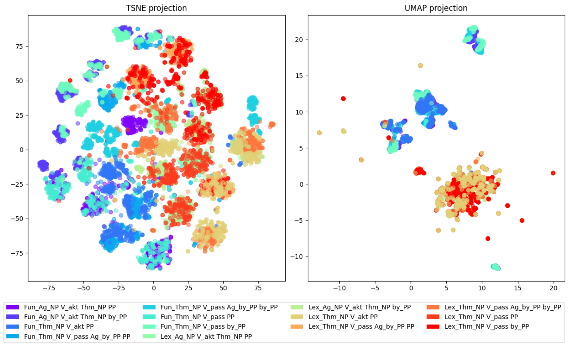

Figure 4: t-SNE and UMAP projections of sentence representations (averaged token embeddings) coloured by their syntactic pattern and the use of lexicalized or functional words.

Syntactic structure and semantic roles represent complex information, which may be encoded by weighted combinations of subsets of dimensions (Bengio et al., 2013; Elhage et al., 2022). We then mine for this information following the approach described in Nastase and Merlo (2024), which uses a variational encoder-decoder to compress sentences into representations that capture syntactic and semantic information. Sentence embeddings from Electra have size 768, and the latent layer in the used system has size 5. To encourage the desired information – in this case syntactic-semantic structure – to be encoded on the latent layer, we form instances by pairing an input sentence s_i with structure str_i with a sentence $s_j \neq s_i$ that has the same structure ($str_j = str_i$), and with N negative examples s_k that have different structures ($str_k \neq str_i$). In our experiments we use N = 7. The structure information is used to build the dataset and obtain a deeper evaluation of the results, but is not provided to the system. We built separate datasets for Fun and Lex.

test on train on	Fun	Lex
Fun	0.976	0.399
Lex	0.487	0.914
Mixed	**0.980**	**0.916**

Table 1: F1 scores on predicting the sentence with the same structure as the input, through a variational encoder-decoder system. For all experiments the system uses 2000 training instances, 10% of which are dynamically selected in each experiment for validation.

This approach enables a two-fold evaluation: (i) in terms of performance in detecting the correct structure, by choosing the candidate answer that has the same syntactic-semantic information as the input; (ii) in terms of the compressed representation on the latent layer, which captures these syntactic and semantic properties.

Table 1 shows the averaged F1 scores over three experiments. We note first that training and testing on the same type (Fun or Lex) leads to high results, thus validating the experimental set-up.

The results on test data of the same type as the training are very different from those on the test of the other type. This indicates that for each of the Fun and Lex data variations, the system discovers different clues to match two sentences with the same structure. The high results when training on the sentences with functional words may also indicate overfitting because of the repetitive vocabulary. Additional information comes from the analysis of the compressed representations on the latent layer, which are expected to capture the sentence structure that is shared by the functional and lexicalized data.

The top two plots of Figure 5 show the projection on the latent layer of the sentence representations with functional and content words, when trained on the sentences with functional words (top) or on the sentences with content words (middle). The plots, matching the F1 scores, show clear clusters for the data that matches the training type, but only slight separation for the data points from the other type.

To test whether there is a shared level of informa-

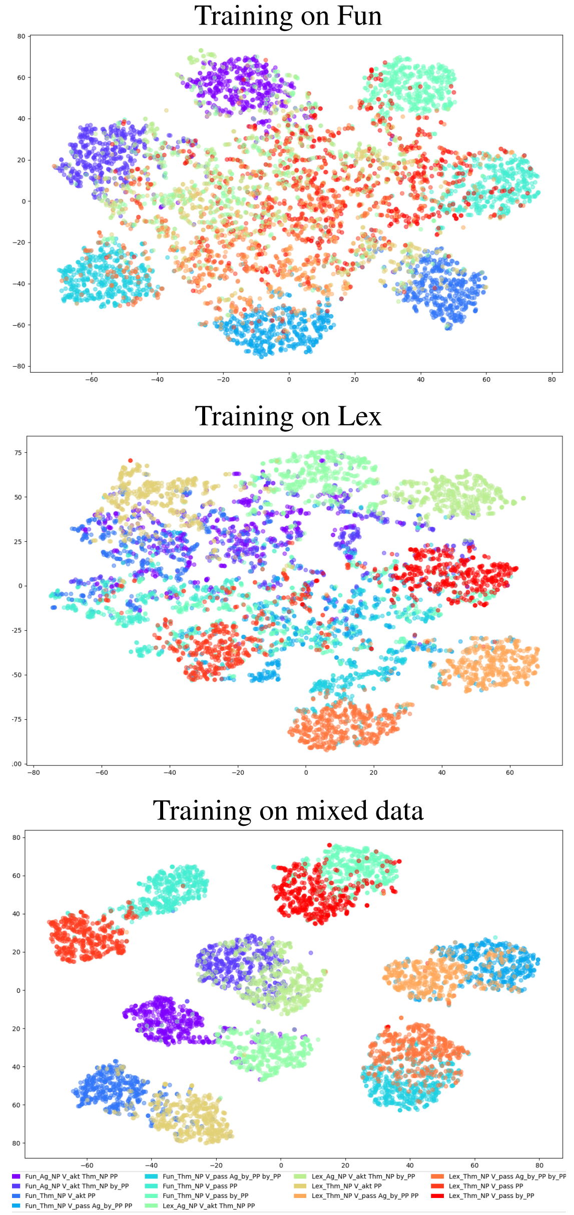

Figure 5: Latent representation analysis: t-SNE projection of vectors on the latent layer for the sentences in the training instances.

tion between sentences with functional or content words, despite what the plots in Figure 4 indicate, we train the system with a dataset containing a mixture of lexicalized and functionalized instances. If there is shared information, we should observe high results on both test sets when training with the mixed training data, *and* overlapping clusters for the compressed representations on the latent layer. If there is no shared information, the results may be high on each test set (because separately they have been very well modelled), but the clusters of the compressed representations would be separate.

The results in Table 1 shows very high results for both datasets for the mixed data training. The analysis of the representations on the latent layer, at the bottom of Figure 5, shows that the system has discovered a shared space between the sentences with functional and those with content words. What these sentences have in common is the syntactic and semantic structure, and the overlapping clusters of the compressed representations on the latent layer confirms that the system has uncovered this shared structure. The overlap between the clusters induced through joint training also supports the idea that the latent layer encodes structure, rather than simply differentiating seven amorphous classes, as there is no overlap between sentences in the Fun and Lex instances.

3.3 Task solving

The previous experiments on the dataset of sentences has shown that shared structure can be detected, but it may be argued that this is a simple clustering due to other types of indicators, and the representation on the latent layer does not actually contain structure and semantic role information. We therefore use the BLM data, to investigate this deeper. The BLM task frames a linguistic phenomenon as a linguistic puzzle. Solving this puzzle relies on detecting the linguistic objects, their relevant properties, and the structure both within each sentence, and across the input sequence. This dataset also allows us to test generalization at several levels, because of the three levels of lexical variation. We report results on the merged COS-OD dataset (see Section 2.4), as it has an added layer of complexity: the correct answer structure for a COS instance is incorrect for OD, and vice-versa. The two classes of verbs exhibit different semantic frames for the intransitive target answers (e.g., patient or agent subjects), with instances of functional and lexical elements tending to align more with either agent or patient roles. This allows us to test whether in contexts with the two classes of verbs, functional and lexicalized phrases encode the necessary properties.

We use the system described by Nastase and Merlo (2024), that solves the BLM problem in two steps: compresses the sentence into a representation that encodes the structure relevant to the BLM puzzle, and uses these compressed representations to solve the multiple-choice puzzle. The system's two steps are encoded through interconnected variational encoder-decoders, as illustrated in Figure 6, which are trained together. The learning objec-

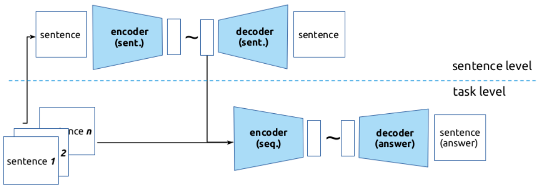

Figure 6: Two-step VAE BLM solver

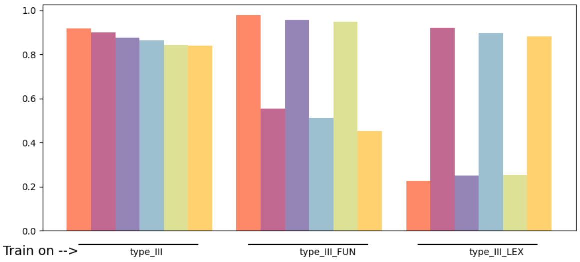

Figure 7: Results in terms of average F1 over three runs for solving the type III (maximal lexical variation) on merged COS and OD BLM tasks. Joint training vs. separate training.

tive is to maximize the score of the correct answer from the candidate answer set, and minimize that of the incorrect ones. During testing, the system constructs the representation of an answer, then chooses the closest one from the given options. All potential answers consist of a verb frame filled with phrases that play specific roles (Section 2). The correct one consists of the combination of phrases whose roles fit together for the given verb, while the other contain similar pieces, but which violate some semantic, syntactic (or both) rules. This setup allows us to test whether specific elements in the sentences from the input sequence, and their semantic roles have been detected and used properly in building the correct answer.

Figure 11 shows the F1 results (as averages over three runs) of joint vs. separate training for the merged COS+OD BLM task[6]. The results are for type III data, with maximum lexical variation. The complete results are in Tables 3 in the appendix.

Processing separately datasets of sentences with and without functional words leads to high results within each task, validating the experimental setup, but leads to low results when testing across tasks. This shows, as in the analysis of the sentences datasets, that for each of the Fun and Lex subsets, the systems discovers and exploits different regularities in the training data. Using a mixed training dataset, instead, encourages all systems to find a shared feature space. This shows that the language model encodes the syntactic-semantic structure of functional and lexicalized parallel sentences in a similar manner.

4 Discussion

The primary goal of this paper is to investigate if LLMs encode lexical abstraction. We have translated this question into properties of words and sentence representations in the embedding space. We have analysed the relative positions of functional and content words in the embedding space, when considered in isolation, or in (syntactic and semantic) structurally parallel sentences, and we have investigated the kind of information encoded in sentence embeddings.

Embeddings of functional words and nouns are mingled in the embedding space, but functional words are also distinguishable from the nouns. This result aligns with the functional words behaviour as place-holders for a wide variety of nouns and in many different contexts, but also having specific characteristics. [7]

Sentence embeddings of parallel sentence variations occupy different regions of the embedding space. This result seems to show that functional and lexical sentence variations do not share much information, as their distance in the embedding space indicates low similarity.

We can detect information about the shared syntactic structure in the embeddings of the functional and lexical variations of the same sentences. Our follow-up experiments uncover information about shared information in Fun and Lex variation of sentences, and of a more complex linguistic puzzle. Training on only one type of data does not reveal the shared syntactic structure and semantic roles, reflecting the shallow differences noted in the sentence embedding space. Training on mixed data, however, leads to high results on both dataset variations *and* overlapping clustered representations in the latent space.

Contrary to our conclusion that the system has discovered a shared space based on the abstraction of nouns, one might argue that the shared space we find is due only to the shared verb, or shared lexical

[6] Results on the separate tasks in Appendix C.

[7] It is interesting to remark, in this respect, that the semantic literature also contains proposals suggesting that pronominal forms are not place-holders, but are better considered as equivalent to noun phrase (NP) descriptions, where they refer to a less abstract, fuller expression in context, in relevant environments (Elbourne, 2002; Lewis, 2022).

material between the train and test partitions. Had that been the case, the cross-testing results, when training on separate data types, would have been closer to the results on mixed data, given that the verb is not replaced by a functional category and it remains the same across all types of data and sentences. This argument is especially true for the type III subset of the BLM task, which has maximal lexical variation.

We think instead that the results indicate that the model trained on the functional data, which has a very small and consistent vocabulary, relies on shallower features, while the model learned on the lexicalized data is more robust, but not sufficiently abstract. Training the system with mixed data leads not only to a model that performs very well on both data variations, but all sentences are projected into the same compressed embedding space, establishing the necessary links between nominal expressions and their functional equivalents that support abstraction and generalisation.

5 Related work

A generalization taxonomy based on an extensive analysis of publications in NLP that deal with the topic of generalization is proposed in Hupkes et al. (2023). They distinguish five main dimensions for generalization analysis: motivation (concerning the higher-level aims of the model), generalization type (the properties of language or domain or model the model is intended to capture), shift type (the kind of differences between training and testing data distributions), shift source (the source of the difference in data distributions) and shift locus (where in the pipeline does the shift in data distributions occurs). This analysis reflects the focus in the NLP community on the model, and its properties from a machine learning point of view.

Language has its own generalization and abstraction dimensions, which could be at the lexical level (Regneri et al., 2024; Sukumaran et al., 2024), concern verb frames (Wilson et al., 2023; Yi et al., 2022), grammar (Kim and Smolensky, 2021) or a combination of these (Wang et al., 2024). The results of such investigations do not reveal a clear picture. While Kim and Smolensky (2021) observe a limited degree of generalization based on grammatical categories, they note that the results may not have been driven by abstraction. Yi et al. (2022) show that both verb and sentence representations encode information about a verb's alternation class, but the linguistic generalization within the verb argument structure is limited, as models fail on unseen contexts. In experiments on an entailment graph that contains abstract concepts entailed by components of events (nouns, verbs, the event as a whole), Wang et al. (2024) show that the LLMs have difficulty understanding abstract knowledge, but they can be improved with fine-tuning.

Structural priming is used in Michaelov et al. (2023) to investigate the degree of grammatical abstraction in LLMs for three verb alternations: active/passive, dative alternation and two forms of possessive. In monolingual and cross-lingual settings, they find evidence for abstract grammatical representations of these phenomena.

Close to the topic of this paper, Regneri et al. (2024) investigate whether hyponymy is encoded in the transformer by analysing the attention matrices when presented with hyponymous noun pairs. In our work, instead, we have analysed the output of a pretrained language model, and whether the word and sentence embeddings it produces encode particular linguistic information that would allow us to establish a parallel between lexicalized and abstract expressions of a sentence.

All this work shows an unclear picture of sentence embeddings, and the information – and its degree of abstractness – it encodes. Our work provides further linguistically-oriented evidence to clarify the relation between embeddings, abstraction and generalisation.

6 Conclusions

Our study contributes to the discussion of generalization in language models, and in particular studies linguistic generalization. We translate the question of whether language models capture lexical abstraction into tests of word and sentence embeddings in the embedding space. We show that the linguistic behaviour of functional words as placeholders for nouns is reflected in the relative positions of their corresponding embeddings. We show that despite the apparent lack of similarity between functional and lexicalized versions of sentences – as shown by their distinct positions in the embedding space – a deeper analysis reveals shared syntactic and semantic information. These conclusions are further reinforced by the results on a problem solving task, the BLM multi-choice problem, whose solution relies on the proper detection of linguistic objects and their relations.

7 Limitations

Synthetic dataset We use a synthetic dataset, for controlled experimentation, which primarily consists of simple sentence structures. The dataset, then, may not fully capture the complexity of language. Future extensions will include many more structures and variations. Another limitation is the all-or-nothing pronominalisation of sentences, where each sentence is either fully categorized into a predefined functional element or not. Future work will have to modulate the amount of pronominalisation and study different patterns of interactions between nominal expressions and their pronominal equivalent. Moreover, at the moment, we do not have comparable results with a human experiment, which could shed light on more human-like abstraction processes. Finally, this study relies exclusively on English data. While many pronominal systems are structured like the one of English, many other pronominal systems exist. Future studies should add a cross-linguistic dimension.

Using encoder transformer models Previous reviewers have commented on the fact that we do not use multiple LLMs for this task, and in particular no generative models. We have justified our choice of the model in Section 3, as Electra is one of the best performing encoder models, and also outperforms XLNet (Yang et al., 2019) and MPNet (Song et al., 2020) on GLUE tasks. The embedding space of different models is different, and we chose to study the embedding space of one of the best performing encoder-based models. For our study we require word embeddings as well as sentence embeddings. While there are ways to elicit approximations of sentence embeddings through word-definition like prompts (Jiang et al., 2024; Zhang et al., 2024), these are only approximations and not direct representations of a target sentence, which is important particularly when the sentences in our data have specific linguistic and grammatical properties, and do not have obvious one word equivalents.

References

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Daniel Büring. 2019. *1. Pronouns*, pages 1–32. De Gruyter Mouton, Berlin, Boston.

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*.

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.

Paul Elbourne. 2002. *Situations and individuals*. Ph.D. thesis, Massachusetts Institute of Technology.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

Martin Haspelmath. 1997. Indefinite pronouns. *Oxford Studies in Typology and Linguistic Theory)/Clarendon Press*.

Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA. Association for Computational Linguistics.

Najoung Kim and Paul Smolensky. 2021. Testing for grammatical category abstraction in neural language models. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 467–470, Online. Association for Computational Linguistics.

Beth Levin. 1993. *English Verb Classes and Alternations A Preliminary Investigation*. University of Chicago Press, Chicago and London.

Karen S Lewis. 2022. Descriptions, pronouns, and uniqueness. *Linguistics and Philosophy*, 45(3):559–617.

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*.

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.

Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408.

James Michaelov, Catherine Arnett, Tyler Chang, and Ben Bergen. 2023. Structural priming demonstrates abstract grammatical representations in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3703–3720, Singapore. Association for Computational Linguistics.

Vivi Nastase and Paola Merlo. 2024. Are there identifiable structural parts in the sentence embedding whole? In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 23–42, Miami, Florida, US. Association for Computational Linguistics.

Michaela Regneri, Alhassan Abdelhalim, and Soeren Laue. 2024. Detecting conceptual abstraction in LLMs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4697–4704, Torino, Italia. ELRA and ICCL.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Priyanka Sukumaran, Conor Houghton, and Nina Kazanina. 2024. Investigating grammatical abstraction in language models using few-shot learning of novel noun gender. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 747–765, St. Julian’s, Malta. Association for Computational Linguistics.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. 2024. Holmes a benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647.

Zhaowei Wang, Haochen Shi, Weiqi Wang, Tianqing Fang, Hongming Zhang, Sehyun Choi, Xin Liu, and Yangqiu Song. 2024. AbsPyramid: Benchmarking the abstraction ability of language models with a unified entailment graph. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3991–4010, Mexico City, Mexico. Association for Computational Linguistics.

Michael Wilson, Jackson Petty, and Robert Frank. 2023. How abstract is linguistic generalization in large language models? experiments with argument structure. *Transactions of the Association for Computational Linguistics*, 11:1377–1395.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

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.

Bowen Zhang, Kehua Chang, and Chunping Li. 2024. Simple techniques for enhancing sentence embeddings in generative language models. In *International Conference on Intelligent Computing*, pages 52–64. Springer.

A Data

Class	Verb
COS	*bake*, *bend*, *blacken*, *break*, *brighten*, *caramelize*, *chip*, *close*, *corrode*, *crinkle*, *defrost*, *empty*, *expand*, *fry*, *harden*, *harmonize*, *heat*, *improve*, *increase*, *intensify*, *melt*, *open*, *propagate*, *purify*, *sharpen*, *shrink*, *sweeten*, *tear*, *whiten*, *widen*.
OD	*clean*, *cook*, *draw*, *drink*, *eat*, *fish*, *hum*, *iron*, *knead*, *knit*, *mend*, *milk*, *nurse*, *paint*, *play*, *plow*, *polish*, *read*, *recite*, *sculpt*, *sew*, *sing*, *sow*, *study*, *sweep*, *teach*, *wash*, *weave*, *whittle*, *write*.

Table 2: Verbs categorized by class

	COSFun - Context		COSFun - Answers
1	She broke it with this	1	**It broke by those there**
2	She broke it by those there	2	She broke by those there
3	It was broken by her with this	3	It was broken by her
4	It was broken by her by those there	4	She was broken by it
5	It was broken with this	5	It broke her
6	It was broken by those there	6	She broke it
7	It broke with this	7	It broke by her
?	???	8	She broke by it

	COSLex - Context		COSLex - Answers
1	The archaeologist broke a vase in the lab	1	**The vase broke by mistake**
2	The archaeologist broke a vase by mistake	2	The archaeologist broke by mistake
3	The vase was broken by the archaeologist in the lab	3	The vase was broken by the archaeologist
4	The vase was broken by the archaeologist by mistake	4	The archaeologist was broken by the vase
5	The vase was broken in the lab	5	The vase broke the archaeologist
6	The vase was broken by mistake	6	The archaeologist broke the vase
7	The vase broke in the lab	7	The vase broke by the archaeologist
?	???	8	The archaeologist broke by the vase

Figure 8: Examples of Fun and Lex for the English verb *break*, one of the verbs belonging to COS class.

	ODLex - Context		ODFun - Answers
1	They paint it with this	1	It painted by that
2	They paint it by that	2	**They painted by that**
3	It was painted by them with this	3	It was painted by them
4	It was painted by them by that	4	They were painted by it
5	It was painted with this	5	It painted them
6	It was painted by that	6	They painted it
7	They painted with this	7	It painted by them
?	???	8	They painted by it

	COSLex - Context		COSLex - Answers
1	These artists paint a portrait with a brush	1	A portrait painted by the lake
2	These artists paint a portrait by the lake	2	**These artists painted by the lake**
3	A portrait was painted by these artists with a brush	3	A portrait was painted by the artists
4	A portrait was painted by these artists by the lake	4	These artists were painted by a portrait
5	A portrait was painted with a brush	5	A portrait painted these artists
6	A portrait was painted by the lake	6	These artists painted a portrait
7	These artists painted with a brush	7	A portrait painted by these artists
?	???	8	These artists painted by a portrait

Figure 9: Examples of Type_I Fun and Lex data for the English verb *paint*, one of the verbs belonging to OD class

B Stand-alone embeddings

Figure 10 shows the t-SNE projection of the word embeddings (as averages over the respective token embeddings) for the functional words and noun phrases in our data, obtained in isolation (when presented to the pretrained model alone). Functional words appear isolated in this space, which indicates that the shared information between the functional elements and the nouns they can replace, should there by any, is not to be found at a shallow level.

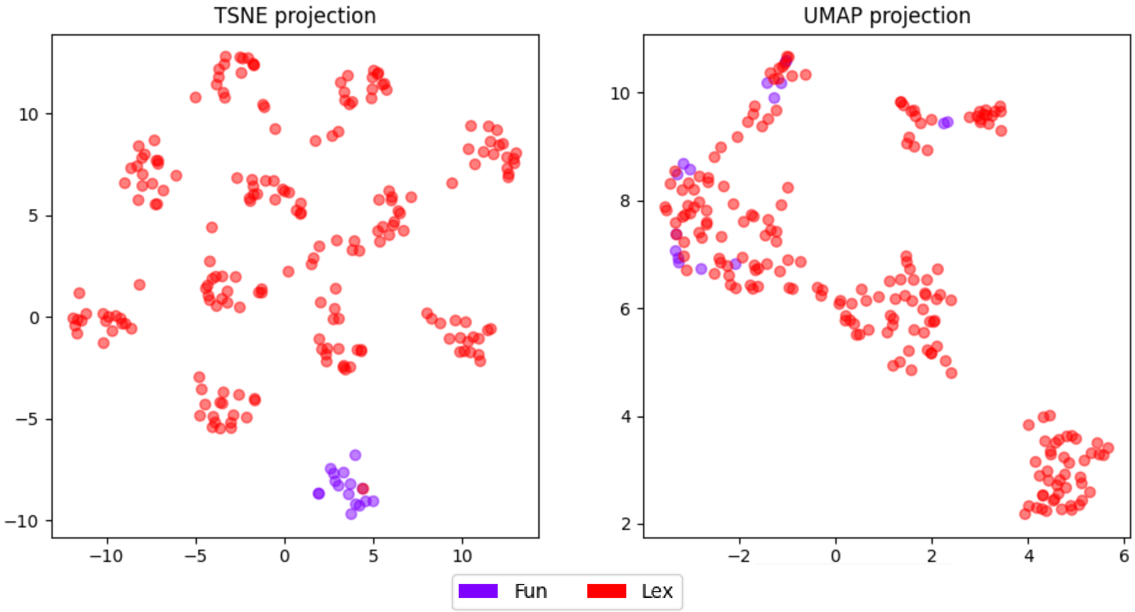

Figure 10: t-SNE and UMAP projections of the embeddings of functional words and nouns, without a sentential context.

C BLM task results

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 systems were trained for 300 epochs, and the results reported are F1 averages (standard deviation) over three runs. The training data for all experiments was 2000 instances. For the joint-training set-up, the data was split evenly across the variations.

test on		**train on**	
		Joint training	
	type_I	type_II	type_III
type_I_Fun	0.866 (0.012)	0.813 (0.039)	0.918 (0.001)
type_I_Lex	0.830 (0.015)	0.784 (0.012)	0.899 (0.002)
type_II_Fun	0.800 (0.013)	0.777 (0.031)	0.877 (0.014)
type_II_Lex	0.773 (0.018)	0.741 (0.006)	0.865 (0.002)
type_III_Fun	0.781 (0.023)	0.791 (0.021)	0.843 (0.010)
type_III_Lex	0.743 (0.005)	0.813 (0.013)	0.838 (0.006)
		Training on Fun	
	type_I_Fun	type_II_Fun	type_III_Fun
type_I_Fun	0.916 (0.007)	0.927 (0.012)	0.979 (0.009)
type_I_Lex	0.440 (0.025)	0.500 (0.022)	0.553 (0.013)
type_II_Fun	0.832 (0.008)	0.891 (0.005)	0.956 (0.011)
type_II_Lex	0.415 (0.021)	0.472 (0.014)	0.511 (0.013)
type_III_Fun	0.849 (0.014)	0.921 (0.002)	0.948 (0.010)
type_III_Lex	0.409 (0.005)	0.445 (0.012)	0.451 (0.003)
		Trainig on Lex	
	type_I_Lex	type_II_Lex	type_III_Lex
type_I_Fun	0.247 (0.009)	0.332 (0.029)	0.227 (0.022)
type_I_Lex	0.803 (0.018)	0.808 (0.016)	0.922 (0.004)
type_II_Fun	0.267 (0.006)	0.339 (0.015)	0.249 (0.038)
type_II_Lex	0.713 (0.007)	0.779 (0.006)	0.896 (0.006)
type_III_Fun	0.297 (0.014)	0.303 (0.009)	0.253 (0.024)
type_III_Lex	0.674 (0.005)	0.847 (0.014)	0.883 (0.005)

Table 3: COS and OD merged tasks: Results as averaged F1 (std) over three runs, for three training set-ups: joint training (training using both Fun and Lex instances), training on Fun instances, training on Lex instances. For all set-ups we use 2000 training instances. For the joint training these are evenly split between Fun and Lex.

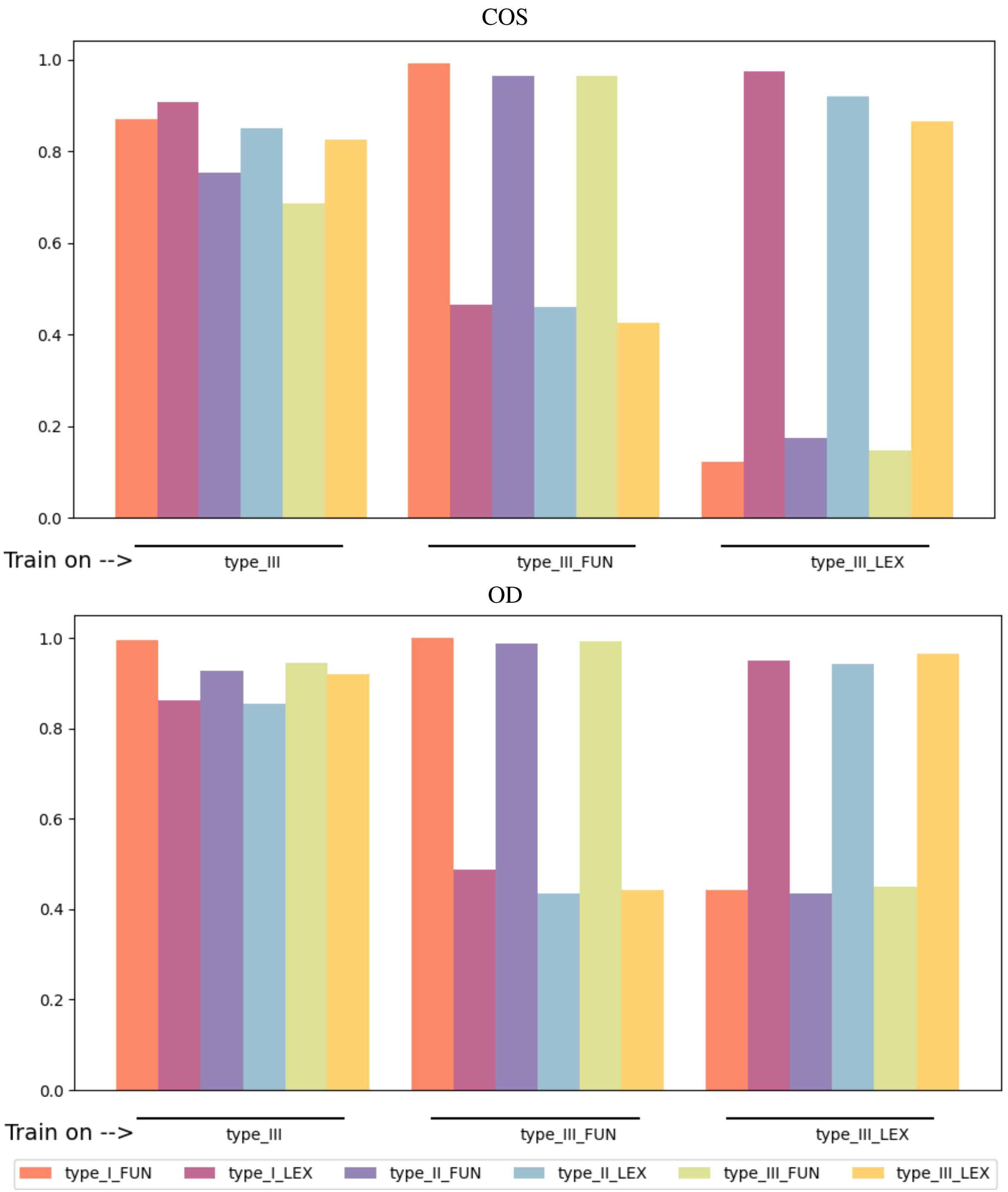

Figure 11: Results in terms of average F1 over three runs for solving the type III (maximal lexical variation) COS and OD BLM tasks for three models. Joint training vs. separate training.

test on		**train on**	
		Joint training	
	type_I	type_II	type_III
type_I_Fun	0.866 (0.010)	0.893 (0.010)	0.871 (0.010)
type_I_Lex	0.861 (0.002)	0.921 (0.013)	0.908 (0.009)
type_II_Fun	0.723 (0.020)	0.812 (0.016)	0.754 (0.010)
type_II_Lex	0.808 (0.006)	0.873 (0)	0.851 (0.002)
type_III_Fun	0.680 (0.009)	0.770 (0.024)	0.686 (0.010)
type_III_Lex	0.790 (0.003)	0.800 (0.021)	0.826 (0.011)
		Training on Fun	
	type_I_Fun	type_II_Fun	type_III_Fun
type_I_Fun	0.981 (0.006)	0.973 (0.003)	0.992 (0.004)
type_I_Lex	0.404 (0.023)	0.433 (0.014)	0.464 (0.006)
type_II_Fun	0.927 (0.018)	0.933 (0.011)	0.964 (0.009)
type_II_Lex	0.396 (0.014)	0.427 (0.012)	0.459 (0.007)
type_III_Fun	0.898 (0.012)	0.933 (0.005)	0.963 (0.003)
type_III_Lex	0.388 (0.024)	0.414 (0.026)	0.426 (0.011)
		Trainig on Lex	
	type_I_Lex	type_II_Lex	type_III_Lex
type_I_Fun	0.160 (0.017)	0.117 (0.005)	0.122 (0.019)
type_I_Lex	0.986 (0.006)	0.960 (0.007)	0.973 (0.005)
type_II_Fun	0.240 (0.005)	0.132 (0.011)	0.174 (0.010)
type_II_Lex	0.877 (0.008)	0.908 (0.008)	0.920 (0.007)
type_III_Fun	0.231 (0.007)	0.150 (0.010)	0.147 (0.005)
type_III_Lex	0.809 (0.006)	0.846 (0.010)	0.864 (0.004)

Table 4: BLM-COS: Results as averaged F1 (std) over three runs, for three training set-ups: joint training (training using both Fun and Lex instances), training on Fun instances, training on Lex instances. For all set-ups we use 2000 training instances. For the joint training these are evenly split between Fun and Lex.

test on	**train on**		
	Joint training		
	type_I	type_II	type_III
type_I_Fun	0.979 (0.006)	0.970 (0.007)	0.994 (0.003)
type_I_Lex	0.683 (0.014)	0.764 (0.016)	0.861 (0.014)
type_II_Fun	0.854 (0.023)	0.898 (0.010)	0.927 (0.005)
type_II_Lex	0.603 (0.021)	0.701 (0.023)	0.854 (0.006)
type_III_Fun	0.886 (0.020)	0.947 (0.020)	0.944 (0.004)
type_III_Lex	0.849 (0.011)	0.882 (0.016)	0.920 (0.007)
	Train on Fun		
	type_I_Fun	type_II_Fun	type_III_Fun
type_I_Fun	1.000 (0)	1.000 (0)	1.000 (0)
type_I_Lex	0.474 (0.011)	0.549 (0.038)	0.489 (0.039)
type_II_Fun	0.923 (0.010)	0.962 (0.008)	0.989 (0.007)
type_II_Lex	0.431 (0.036)	0.476 (0.026)	0.436 (0.022)
type_III_Fun	0.942 (0.014)	0.980 (0.005)	0.993 (0.003)
type_III_Lex	0.365 (0.015)	0.487 (0.017)	0.443 (0.012)
	Training on Lex		
	type_I_Lex	type_II_Lex	type_III_Lex
type_I_Fun	0.489 (0.037)	0.480 (0.007)	0.442 (0.020)
type_I_Lex	0.746 (0.007)	0.796 (0.015)	0.951 (0.013)
type_II_Fun	0.43 (0.005)	0.467 (0.005)	0.436 (0.018)
type_II_Lex	0.682 (0.026)	0.712 (0.010)	0.943 (0.014)
type_III_Fun	0.501 (0.008)	0.466 (0.011)	0.451 (0.014)
type_III_Lex	0.876 (0.017)	0.899 (0.008)	0.964 (0.009)

Table 5: BLM-OD: Results as averaged F1 (std) over three runs, for three training set-ups: joint training (training using both Fun and Lex instances), training on Fun instances, training on Lex instances. For all set-ups we use 2000 training instances. For the joint training these are evenly split between Fun and Lex.