The Typology of Ellipsis: A Corpus for Linguistic Analysis and Machine Learning Applications

Damir Cavar
Indiana University
dcavar@iu.edu

Ludovic Vetea Mompelat
University of Miami
lvm861@miami.edu

Muhammad Abdo
Indiana University
mabdo@iu.edu

Abstract

Ellipsis constructions are challenging for State-of-the-art (SotA) Natural Language Processing (NLP) technologies. Although theoretically well-documented and understood, there needs to be more sufficient cross-linguistic language resources to document, study, and ultimately engineer NLP solutions that can adequately provide analyses for ellipsis constructions. This article describes the typological data set on ellipsis that we created for currently seventeen languages. We demonstrate how SotA parsers based on a variety of syntactic frameworks fail to parse sentences with ellipsis, and in fact, probabilistic, neural, and Large Language Models (LLM) do so, too. We discuss experiments that focus on detecting sentences with ellipsis, predicting the position of elided elements, and predicting elided surface forms in the appropriate positions. We show that cross-linguistic variation of ellipsis-related phenomena has different consequences for the architecture of NLP systems.

1 Introduction

Ellipsis is a linguistic phenomenon that results in the omission of words in sentences that are usually obligatory in a given syntactic context and that the speaker and hearer can understand and reconstruct without effort. Simple noun phrase (NP) or Forward Conjunct Reduction (FCR), as in example (1), is common cross-linguistically.

(1) a. My sister lives in Utrecht and ___ works in Amsterdam.
b. My sister lives in Utrecht and **she/my sister** works in Amsterdam.

The possibility to elide phrases or words in coordinated constructions has universal and language-specific aspects to it. Common FCR is possible in all languages we are aware of. It is not only possible but the preferred form of presentation in text or spoken language whenever coordination occurs. If ellipsis can be applied in unmarked cases, it is applied. The form in (1b) without ellipsis might be perceived as emphatic or, in a pragmatic or semantic sense, as specific, in contrast to the unmarked default in example (1a).

Other variants of ellipsis include so-called *gapping*, as in (2a) where the verb complex *is reading* is elided. In example (2b), a case of VP-Ellipsis, the entire predicate or Verb Phrase (VP) is elided.

(2) a. Peter is reading a book and Mary ___ a newspaper.
b. She will hi-five Daniel, but I won't ___

Such ellipsis phenomena are context-independent and intra-sentential because no context outside of the sentence boundaries is necessary to license the ellipsis.

Context-dependent forms of ellipsis can be found in responses to questions, as in example (3). In the response (3b), the words *each candidate will talk* are elided.

(3) a. Will each candidate talk about taxes?
b. No, ___ about foreign policy.

While English exhibits limited examples with lexical mismatches of elided word forms, as in example (4a), highly inflecting languages like Hindi or Croatian (4b) show that the elided words do not have to be homophonous. The words in round brackets in (4) are preferably elided in unmarked contexts.

(4) a. John **reads** a book, but Paul and Mary (**read**) a newspaper.
b. Ivan **je čitao** knjigu a Marija i Petar (**su čitali**) novine.
I. be read book but M. and P. be read newspaper

Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP (SIGTYP 2024), pages 46–54
March 22, 2024 ©2024 Association for Computational Linguistics

Elided elements can also be scattered over multiple positions in a clause, as in example (5), where the words *will*, *greet*, and *first* are elided in the respective slots in the second conjunct.

(5) Will Jimmy greet Jill first, or ___ Jill ___ Jimmy ___ ?

As discussed in Testa et al. (2023) and Hardt (2023), ellipsis constructions are very common and often accompanied by specific semantic effects. While various quantifier scope effects[1] can be observed in ellipsis constructions, Common semantic issues involve so-called *zeugma* (Sennet, 2016) effects as in example (6).

(6) a. John stole a book and Peter stole kisses from Mary.

b. John stole a book and Peter ___ kisses from Mary.

The second conjunct in example (6a) includes an idiomatic predicate causing semantic deviation without significantly impacting grammaticality judgments.

We observed that NLP pipelines fail to provide appropriate syntactic structures for such sentences in downstream tasks and for common information extraction from business reports or medical documents. Using ellipsis constructions from our corpus, we tested the most recent versions of Stanza (Qi et al., 2020), spaCy (Honnibal and Johnson, 2015), Benepar (Kitaev and Klein, 2018; Kitaev et al., 2019), and the Xerox Linguistic Environment (XLE) (Crouch et al., 2011). None of the parse trees based on the different grammar formalisms were adequate in our evaluation. Our team of syntacticians judged the adequacy of parse trees. This is true for SotA Dependency parsers, neural Constituency parsers, as well as for rule-based systems like the XLE-based Lexical-functional Grammar (LFG) parser using the English, German, or Polish grammar. LLMs are as challenged with such constructions as these rule-based, statistical, or neural syntactic parsers.

Figure 1 shows an example in which the overt subject of the first conjunct is labeled as the subject. The same element is the syntactic subject and semantic object of the second conjunct. These functional relations are missing in the Dependency tree and cannot be easily resolved in a generic way for any kind of ellipsis construction with multiple conjoined clauses. While this parse tree might be argued to result from the Dependency Grammar framework as such, all parse trees that we have analyzed were definitely useless for subsequent information retrieval or semantic analysis that depend on sentential functional relations of clausal constituents, as for example, the arguments subject and object of the main predicate in the clause.

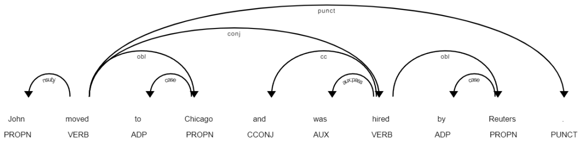

Figure 1: Stanza Dependency Tree.

Additional errors emerge when looking at simple gapping constructions as in Figure 2. While in most cases, the parser would generate a hypothesis that indicates that *bicycle* and *Mary* are coordinated, in this case, the parser coordinates the verb of the first conjunct and the object noun, declaring *Mary* to be the subject of the nominal object *truck*.

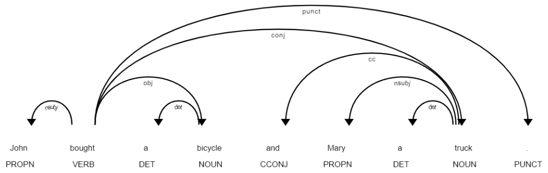

Figure 2: Stanza Dependency Tree.

This does not improve when looking at constituency parser outputs, as in Figure 3. The constituency parser assumes the coordination to be local, rather than clausal, that is, *a bicycle and Mary* is analyzed as the object of *buying*, and the Noun Phrase *a truck* appears to be an orphaned object of the same predicate, too.

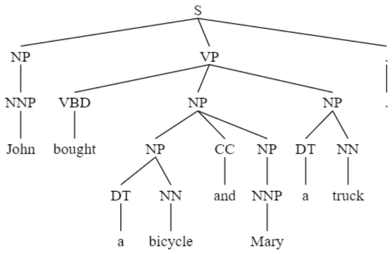

Figure 3: Stanza Constituency tree

For a simple gapping construction XLE using the English grammar generates a C-structure as in

[1] A discussion of semantic changes caused by quantifier scope effects in ellipsis constructions would go beyond the scope of this article.

Figure 4 corresponding to the constituency parse tree in Figure 3.

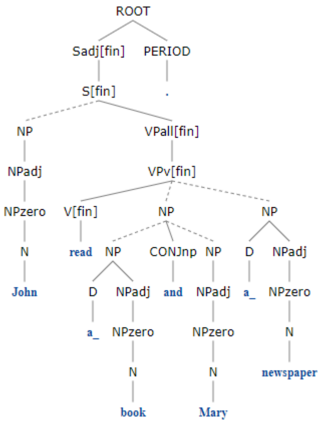

Figure 4: XLE English C-structure

In LFG the F(functional) structure represents morphosyntactic features of c-structure phrases and lexical elements, as well as grammatical functions like subject and object. The corresponding F-structure in Figure 5 provided by XLE shows that the coordination is wrongly assumed to be local, implying that *John* engaged in reading *a book and Mary*.

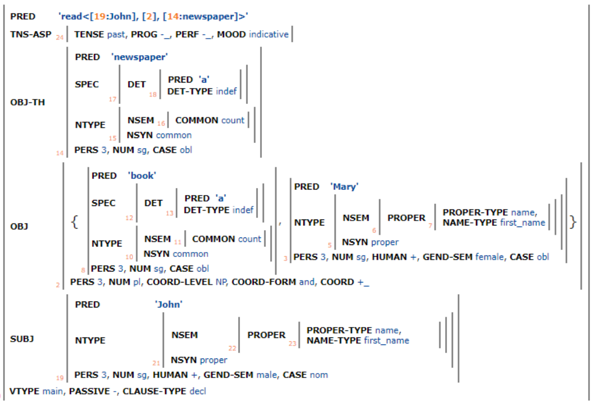
PRED 'read<[19:John], [2], [14:newspaper]>'
TNS-ASP 24 TENSE past, PROG -_, PERF -_, MOOD indicative
OBJ-TH
PRED 'newspaper'
SPEC 17 DET 18 PRED 'a' DET-TYPE indef
NTYPE 15 NSEM 16 COMMON count
NSYN common
14 PERS 3, NUM sg, CASE obl
OBJ
PRED 'book'
SPEC 12 DET 13 PRED 'a' DET-TYPE indef
NTYPE 10 NSEM 11 COMMON count
NSYN common
8 PERS 3, NUM sg, CASE obl
PRED 'Mary'
NTYPE 5 NSEM 6 PROPER 7 PROPER-TYPE name, NAME-TYPE first_name
NSYN proper
3 PERS 3, NUM sg, HUMAN +, GEND-SEM female, CASE obl
2 PERS 3, NUM pl, COORD-LEVEL NP, COORD-FORM and, COORD +_
SUBJ
PRED 'John'
NTYPE 21 NSEM 22 PROPER 23 PROPER-TYPE name, NAME-TYPE first_name
NSYN proper
19 PERS 3, NUM sg, HUMAN +, GEND-SEM male, CASE nom
VTYPE main, PASSIVE -, CLAUSE-TYPE decl

Figure 5: XLE English F-structure

These examples are not rare mistakes that these parsers make in constructions with ellipsis. These are the typical mistakes that we observe in the vast majority of ellipsis constructions.

The following data and corpus creation and experiments were motivated by the fact that document types like business reports, medical or technical documentation, as well as social media content, chat, or spoken language discourse, contain a large number of sentences with ellipses. Given that common SotA NLP pipelines fail to provide adequate syntactic representations as tree structures, higher-level processing of discourse and semantic properties is not possible using their output.

Our motivation to create larger data sets for a larger group of languages was not only driven by this fact but also by a lack of a comparative typological description of ellipses in different languages and across different language groups.

As the example in (4) shows, morphologically rich languages allow lexically matching words to be elided, although the morpho-phonological surface form does not match. This does not seem to be a challenge for native speakers of these languages. However, it is a significant computational challenge to identify the correct morpho-phonological forms that were subject to ellipsis.

Scattered ellipsis, as in example (5), does not appear to be cognitively challenging, either; however, from a Machine Learning (ML) and NLP perspective, we expect to see significant errors and issues in identifying the ellipsis slots and guessing the elided words.

Other typologically interesting aspects of ellipsis and cross-linguistic comparisons are related to the unmarked underlying word order. While VP-ellipsis might manifest itself in different ways in SVO languages, it might result in very different surface phenomena in SOV languages like Hindi or German. In the German example (7) the VP containing the direct object and main verb (*Mutter helfen*) can be elided in the first conjunct.

(7) Karl soll seiner (**Mutter helfen**) und Maria soll ihrer Mutter helfen.

While we have a good understanding of VP-ellipsis in English, it must be made clear whether an elided verb in a transitive predicate construction in Hindi is similar to gapping or, rather, the result of partial VP-ellipsis.

There are numerous research questions that we try to address. On the one hand, syntax internal constraints license sentence-internal ellipsis. In gapping constructions, the gapped verb is not necessarily licensed by discourse conditions and previously mentioned context. On the other hand, the ellipsis of discourse-introduced and -linked words and phrases cannot be assumed to be restricted by purely syntactic constraints. At the same time, complex gapping constructions seem to indicate that ellipsis is not restricted by syntactic phrase boundaries, but rather licensed by phonological correspondence of word sequences. As example 8 shows, the repeated word sequence can be elided, ignoring syntactic phrase structure boundaries.

(8) Jimmy was always dreaming about going to Paris, and Mary ___ to Tokyo?

One interpretation of 8, the default one, implies that *Mary was always dreaming about going to Tokyo*. The elided sequence of words, in this case, does not match with clear-cut syntactic phrase boundaries.

The corpus and research presented in this article are part of the Hoosier Ellipsis Corpus (HEC) project at the NLP-Lab. The goal of the HEC Project is to provide a resource for qualitative and quantitative typological studies of ellipsis over different language types and groups, as well as to provide corpora for the evaluation and development of NLP pipelines that can generate semantically more adequate syntactic structures for ellipsis constructions.

1.1 Previous Work

There is a rich body of literature covering ellipsis in linguistics, as summarized in the Handbook of Ellipsis (van Craenenbroeck and Temmerman, 2018). Summarizing the data discussed and the different theoretical approaches presented in these articles would go beyond the scope of this article. In the following, we focus on the most recent computational approaches and descriptions of ellipsis corpora.

Testa et al. (2023) built a dataset of elliptical constructions, *ELLie*, and evaluated GPT-2 (Radford et al., 2019) and BERT (Devlin et al., 2018), two Transformer-based language models, on their ability to retrieve the omitted verb in elliptical constructions that demonstrate the impact of prototypicality and semantic compatibility between the missing element and its arguments. They found that while the performances of the two language models were influenced by the semantic compatibility of an elided element and its argument, these models had an overall limited mastery of elliptical constructions.

Anand et al. (2021) built the Santa Cruz sluicing dataset. In sluicing constructions as in (9) the elided word list (*John/he can play*) is preceded by an interrogative pronoun (*what*).

(9) John can play something, but I don't know what (**he can play**).

They compiled a corpus of 4,700 instances of sluicing in English, with each instance represented as a short text and annotated for syntactic, semantic, and pragmatic attributes. Most of the data they used comes from the New York Times subcorpus of the English Gigaword corpus. The data set was created by identifying all verb phrases whose final child was a wh-phrase and then manually culling false positives. Each of the instances is marked with five tags, namely, the antecedent, the wh-remnant, the omitted content, the primary predicate of the antecedent clause, and the correlate of the wh-remnant, if available.

Motivated by the assumption that noun ellipsis is more frequent in conversational settings, Khullar et al. (2020) compiled NoEl (An Annotated Corpus for Noun Ellipsis in English), where they annotated the first 100 movies of the Cornell Movie Dialogs dataset for noun ellipsis. Their annotation process involved using the Brat annotation tool to mark ellipsis remnants and their antecedents in the dataset. The dataset was manually annotated by three linguists, and an inter-annotator agreement was measured using Fleiss's Kappa coefficient, which indicated a high level of agreement among annotators. Their results show that a total of 946 cases of noun ellipsis existed in their corpus, corresponding to a rate of 14.08 per 10,000 tokens. The models they used included Naive Bayes, Liner and RBF SVMs, Nearest Neighbors, and Random Forest. They achieved an F1 score of 0.73 in detecting noun ellipsis using linear SVM and 0.74 in noun ellipsis resolution using Random Forest.

Droganova et al. (2018a,b) first created treebanks containing elliptical constructions for English, Czech, and Finnish, using the Universal Dependencies (UD) (Nivre et al., 2016) annotation standard by artificially introducing ellipsis to the sentences. They evaluated several parsers in order to identify typical errors these parsers generate when dealing with elliptical constructions. Note that UD v2 used the *orphan* relation to attach the orphaned arguments to the position of the omitted element. The authors found that the F1-scores of most parsers were below 30%. This highlights how difficult it is for dependency parsers to identify elliptical constructions and warrants data enrichment for ellipsis resolution to improve dependency parsers' performances.

Liu et al. (2016) investigated Verb Phrase Ellipsis (VPE) and conducted three tasks on two datasets. The first dataset consists of the Wall Street

Journal (WSJ) section of the Penn Treenbank with VPE annotation (Bos and Spenader, 2011), and the second dataset was compiled from the sections of the British National Corpus annotated by Nielsen (2005) and converted by Liu et al. (2016) to the format used by Bos and Spenader (2011). The first task consisted of identifying the position of the element, called *target*, that is used to represent the elided verb phrase, called the *antecedent*. This first task only treats cases in which such a *target* is overtly present in the case of VPE, but this is not always the case, as shown in example 2b. The second and third tasks consisted of correctly linking the *target* to its *antecedent* and identifying the exact boundaries of the *antecedent*. Liu et al. (2016) found that the second and third tasks yielded better results when they were treated separately using two different learning paradigms rather than when they were treated jointly. They also found that a logistic regression classification model worked better for the first and third task, but that a ranking-based model yielded better results for the second task.

McShane and Babkin (2016) developed ViPER (VP Ellipsis Resolver), which is a system that uses linguistic principles, and more specifically syntactic features, to detect and resolve VP ellipsis. This system is knowledge-based and does not use empirical data for training. It is not intended to solve all cases of VP-ellipsis, and instead, it first detects the cases of VP ellipsis that are simple enough for the system to treat and then uses string-based resolution strategies. The system identifies the best string to fill and replace the elliptical gap (*sponsor*). The system, evaluated against a GOLD standard dataset generated by the authors, had correctly resolved 61% of the VP ellipsis constructions it identified as simple enough to treat from the Gigaword corpus.

1.2 Summary

The previous work described above was mainly focusing on isolated ellipsis types or specific languages. Our goal was to build on the previous work and expand the data set to more languages and language types, and to broaden the ellipses types documented and studied to the full known set of constructions.

2 The Hoosier Ellipsis Corpus

The Hoosier Ellipsis Corpus V 0.1 consists of data from seventeen languages. Among those languages are low-resourced languages like Navajo, a language of the Athabaskan branch of the Na-Dené language family, and Kumaoni, an Indo-Aryan language spoken in northern India and parts of western Nepal, as well as common Slavic languages (Russian, Ukrainian, Polish), Germanic languages (English, German, Swedish), as well as Hindi, Arabic, Japanese, and Korean.

The corpus includes the following ellipsis types: VP-ellipsis, Sluicing, Gapping, Stripping, Forward (FCR), and Backward Coordinate Reduction (BCR).

The collected data set consists of sentence pairs and possible contexts that precede or follow the target sentence in a text or discourse. The examples in the corpus are collected from linguistic and typological literature. Example sentences from low-resourced languages were collected and validated by native speakers.

We selected a simple Unicode text-based format to encode the data using separator lines and line prefixes to indicate the data entry type. In the encoded data files, the target sentence with an ellipsis is followed by a line of 4 dashes. Within the ellipsis target structure, three underscores indicate each canonical position of the elided word sequence. Complex ellipsis constructions can contain numerous elided slots. The pair of sentences with ellipses and the full form are optionally accompanied by meta information indicated by lines that start with the hash symbol. In the meta-information lines we provide the opportunity to translate the sentence, to provide the original source of the example from publications, and to specify who contributed this example to the data collection. Each example sentence in the data file is followed by at least one empty line. Figure 6 shows a sample entry with the core elements.

```
Wird sie kommen oder ___ er gehen?
----
Wird sie kommen oder wird er gehen?
# TR eng: Will she come or will he go?
# added by: John Smith
# source: Wolfgang Klein (1981)
#         Some Rules of Regular ...
```

Figure 6: Example entry in the Ellipsis Corpus.

This annotation format allows us to indicate and study the distribution of elided elements in the clause. It also provides the 'understood' or 'implied' sequence of words as understood by human native speakers. From a computational perspective, this format allows us to train models that detect the positions of elided elements in sentences. We can also train models that generate the elided word forms. We can use the data set to evaluate existing models and, in particular, LLMs, as discussed in the following section.

The format allows us to convert most of the ellipsis and full-form pairs into the UD 2 format for encoding ellipsis.[2] At the same time, tree structures based on the different grammar formalisms can be encoded as bracketed-notation strings, triple sets for dependencies, or c- and f-structure strings in the meta-information section of each example.

The Ellipsis Corpus is continuously expanded. Many languages in the corpus are expanded using examples from peer-reviewed publications and theoretical or documentary linguistics publications. For low-resourced languages, we rely on contributions from native speakers and their speaker communities. While some of the languages are as of writing this article under-represented, we describe the following experiments and results for a couple of languages that we collected sufficient data on for training models and evaluating their performance, or testing the performance of pre-trained LLMs.

3 NLP Experiments: Methods & Results

We designed three main experimental settings to test the capabilities of current SotA NLP technologies. The tasks are described as follows:

1. Detection of ellipsis in sentences as a general binary classification task.

2. Identification of the positions of elided words or phrases in sentences with ellipsis.

3. Prediction of the correct surface form (morpho-phonological shape) of elided words in sentences with ellipsis.

These tasks we compare across three different NLP-approaches:

1. Logistic Regression classifier

2. Transformer-based classifier and labeler

3. Large Language Models

We assume that the Logistic Regression approach represents a baseline for the binary classification task but that it is less useful for guessing the positions of elided words or generating the elided word forms.

While we expected transformer-based models to perform well as classifiers, we also expected that they would be less efficient for guessing the position of elided elements.

We expected current SotA LLMs to be most successful in all three tasks, in particular when it comes to the generation of the elided word forms since this is the natural task for Generative AI models.

3.1 Dataset

Using our manually compiled Ellipsis Corpus, we constructed three datasets. For English, we expanded the data with the ELLie corpus Testa et al. (2023). We added some corrections and modifications to the ELLie corpus since some native speakers complained about the naturalness of some sentences. We also used sluicing examples from the Santa Cruz Sluicing dataset (Anand et al., 2018).

The first dataset was aimed at a simple binary classification task to detect and label sentences with 1 if they contain ellipsis and with 0 if not. The binary classification datasets were monolingual and a balanced mixture of target sentences and distractors. We generated a 10-fold randomized rotation of the examples to minimize any kind of sequencing effect when training classifiers or

Our corpus comprises pairs of examples showcasing ellipsis constructions, which specify both the location of the omitted element and the full form.

At this early stage of the Ellipsis Corpus, the languages that were represented with sufficient data were English, Russian, Arabic, and Spanish. The experiments described in the following thus focus on these languages. We limit our description here to English and Arabic, since the format and results are equivalent to the settings for the other languages.

3.1.1 English Data

For English, we used 575 examples from ELLie and 559 examples from our manually compiled English Ellipsis Sub-Corpus. Combining each of the datasets with 658 distractor sentences, we generated a ten-fold randomized rotation of sentences.

[2] See for details https://universaldependencies.org/u/overview/specific-syntax.html.

For Task 1, the classification of ellipsis, we generated sentence and label tuples using the label 1 for ellipsis and 0 for no ellipsis.

For Task 2, we generated pairs of ellipsis and full-form sentences, leaving the underscore indicators in the ellipsis example sentence to be able to train labeling algorithms that predict the ellipsis position or to evaluate predicted ellipsis positions directly.

3.1.2 Arabic Data

For the experiments on Modern Standard Arabic, we selected 375 target structures that contain ellipses from the manually compiled Arabic Ellipsis Sub-Corpus and combined those sentences with 500 distractor sentences. The distractor sentences were a random selection of examples without ellipses, as well as the full-form sentences from the ellipses corpus. To the best of our knowledge, there is no other such corpus of Ellipsis in Arabic. The Arabic Ellipsis Sub-Corpus covers various types of syntactic ellipsis (e.g., NP ellipsis, VP-ellipsis, gapping, fragment answers, forward and backward coordinate reduction, and sluicing as in example 10).

(10) لا بد من منع مثل هذه الكارثة ولكن كيف ___؟
[We have to stop this crisis but how___?]

3.2 Task 1: Binary Sentence Classification

The goal of task 1 was to evaluate the performance of baseline approaches with transformer models and LLMs. As the baseline approach, we specified a simple Logistic Regression (LR) model that uses a sentence vectorization approach based on ten simple cues using linguistic intuition. For the generation of cue vectors for each sentence, we used the spaCy[3] NLP pipeline with the part-of-speech tagger and Dependency parser. The classification vectors for each English sentence were generated using the following information:

the number of nouns
the number of subject dependency labels
the number of object dependency labels
the number of conjunctions
the number of *do so*
a boolean whether a *wh*-word is sentence-final
the number of verbs
the number of auxiliaries
the number of *acomp* Dependency labels
the number of tokens *too*

We trained a binary LR classifier using these ten-dimensional vectors. The goal was not to optimize the classifier and achieve the best possible result but to develop a simple baseline classifier using just a few linguistic cues for ellipsis constructions.

The transformer-based classifier is based on BERT for English and the language-specific counterparts for the other languages.

3.3 Task 2: Locate of Ellipsis

In this task, we evaluate Language Models and specific transformer models with respect to their ability to predict the precise location of elided words. The complexity in this task varies from one elided word, multiple elided words as in example (8), and scattered multi-slot ellipsis as in example (5).

The data set for this task consists of sentence pairs. One sentence contains the indicators (3 underscores) for the ellipsis positions, while the other one does not contain such indications and is used for testing the models. The models are trained and tested only using examples that contain ellipses.

Ten-fold random rotations of examples are tested on BERT-based sequence labeling and GPT-4.

For GPT-4 we used a prompt with a rich context: "Annotate the following sentence by placing ___ in the position of each ellipsis. Ellipses indicate gapping, pseudogapping, stripping, and sluicing. If there are no ellipses, answer with only the original sentence."

We have not run few-shot experiments for task 2 yet. but will report on those in the near future.

3.4 Task 3: Generate Elided Words

In this task, we evaluate LLMs for their ability to generate the elided word in the correct positions. The data set consists of sentence pairs. One of the sentences contains ellipsis and the other is the "full-form" of the same sentence with the elided words spelled out. Only examples with ellipses were used for training and testing the models.

For the GPT-4-based evaluation, we used a prompt with a rich context: "Insert any missing words implied by ellipses. Ellipses indicate gapping, pseudogapping, stripping, and sluicing. Answer with only the new sentence. If there are no ellipses, answer with only the original sentence."

As for task 2, we have not performed few-shot experiments for task 3 yet, leaving these experiments for future work.

[3] See https://spacy.io/ for more details.

4 Results

In the zero-shot GPT-4 setting, we used the context "You are a linguistic expert." The prompt "Classify the following sentence as containing ellipsis or not and return a 1 for a sentence with ellipsis and a 0 for a sentence without ellipsis" was preceding each sentence.

We tested various LLMs, including GPT-4, GPT-3.5, Falcon, Llama, Zephyr. We decided to focus on GPT-4 only, since none of the other LLMs turned out to be useful in any of the three tasks, given accuracies below 0.5. For task 1 for English the results are given in table 1. The results for languages like Arabic, Russian, or Spanish vary insignificantly.

model	**accuracy**
LR	0.74
BERT	**0.94**
GPT-4 zero-shot	0.72

Table 1: Task 1 Binary Classifier

It is surprising that the GPT-4 zero-shot classification is worse than the LR-baseline, and significantly worse than the BERT-based classifier. The precise scores from the zero-shot GPT-4 experiment show a Recall of 0.599, a Precision of 0.756, and an F1 Score of 0.668.

In task 1, the zero-shot GPT-4 experiment achieved using the Arabic data resulted in a surprising accuracy of 0.87.

In the default setting, the output from GPT-4 is not discrete for a given example sentence. With the temperature set to the default 0.7 when requesting a label for a sentence in task 1, 2 of 10 responses were the opposite label. When we reduce the temperature to 0, there are no mismatches; the judgments were deterministic. However, with the temperature set to 0, the accuracy was 70%. With the temperature set to 0.7, the accuracy was 75%.

In task 2, we tested an initial BERT-based ellipsis position guesser and achieved first test accuracy of 0.7. The GPT-4-based experiments on task 2 were challenging. The prompt engineering for the zero-shot experiment resulted in an accuracy of only 0.15 for the English data.

For task 3, we exclusively focused on the evaluation of GPT-4. In this task, using the zero-shot strategy, we achieved an accuracy of 0.25 with GPT-4.

5 Conclusion

Ellipsis constructions are obviously still challenging for all the common SotA NLP pipelines, including rule-based systems like the LFG-based XLE. Use of Dependency or Constituency parse trees, or even LFG c- and f-structures for syntactic and semantic processing of real-world data from different genres or registers is limited due to the fact that ellipsis is a common and widespread phenomenon in all languages.

The problem can be partially linked to grammar frameworks like Dependency Grammar or LFG, which do not necessarily foresee opaque linguistic elements (e.g., elided words or phrases) to be active rule elements modeled in grammar rules or descriptive formal annotation frameworks. While UD provides the instruments for annotating or handling ellipses, those instruments need to be more extensive to describe the different intra- and cross-linguistic ellipses types. We also suspect that parsing algorithms and the training of parsers need to include such opaque elements and potentially new learning strategies.

The fact that specific models trained on the prediction of ellipses in sentences outperform LLMs seems to indicate that the lack of explicit data and pure self-supervised machine learning is not sufficient to handle opaque elements in language, either. Training LLMs on purely overt data ignores significant properties of language. Ellipsis phenomena are grammatical and systematic, and it seems problematic for current LLMs to guess covert continuations.

Given that there is too little data on ellipsis in general, and none at all for most languages, it seems necessary to continue our Ellipsis Corpus project and provide not only sufficient data for the different languages, but also a good typological overview of the different manifestations of ellipsis phenomena in different languages and language groups.

Acknowledgements

We are grateful to the NLP-Lab team, in particular Luis Abrego, Billy Dickson, Vance Holthenrichs, Calvin Josenhans, Soyoung Kim, John MacIntosh Phillips, Zoran Tiganj, Khai Willard, Yuchen Yang.

The Ellipsis Corpus and the relevant code for the experiments described in the article are made accessible on GitHub (https://github.com/dcavar/hoosierellipsiscorpus).

References

Pranav Anand, Daniel Hardt, and James McCloskey. 2021. The Santa Cruz sluicing data set. *Language*, 97(1):e68–e88.

Pranav Anand, Jim McCloskey, and Dan Hardt. 2018. Santa Cruz Ellipsis Consortium Sluicing Dataset (1.0).

Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of VP ellipsis. *Language resources and evaluation*, 45:463–494.

Richard Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell, III, and Paula Newman. 2011. *XLE Documentation*. Xerox Palo Alto Research Center, Palo Alto, CA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kira Droganova, Filip Ginter, Jenna Kanerva, and Daniel Zeman. 2018a. Mind the gap: Data enrichment in dependency parsing of elliptical constructions. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 47–54.

Kira Droganova, Daniel Zeman, Jenna Kanerva, and Filip Ginter. 2018b. Parse me if you can: Artificial treebanks for parsing experiments on elliptical constructions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Daniel Hardt. 2023. Ellipsis-dependent reasoning: a new challenge for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 39–47, Toronto, Canada. Association for Computational Linguistics.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Payal Khullar, Kushal Majmundar, and Manish Shrivastava. 2020. NoEl: An annotated corpus for noun ellipsis in English. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 34–43.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Wolfgang Klein. 1981. Some rules of regular ellipsis in German. In W. Klein and W.J.M. Levelt, editors, *Crossing the Boundaries in Linguistics. Studies Presented to Manfred Bierwisch*, pages 51–78. Reidel, Dordrecht.

Zhengzhong Liu, Edgar Gonzàlez, and Dan Gillick. 2016. Exploring the steps of verb phrase ellipsis. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016), co-located with NAACL 2016*, pages 32–40, San Diego, California. Association for Computational Linguistics.

Marjorie McShane and Petr Babkin. 2016. Detection and resolution of verb phrase ellipsis. *Linguistic Issues in Language Technology*, 13.

Leif Arda Nielsen. 2005. *A corpus-based study of verb phrase ellipsis identification and resolution*. Ph.D. thesis, Citeseer.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Adam Sennet. 2016. Polysemy. In S. Goldberg, editor, *Oxford Handbooks Online: Philosophy*. Oxford University Press.

Davide Testa, Emmanuele Chersoni, and Alessandro Lenci. 2023. We understand elliptical sentences, and language models should too: A new dataset for studying ellipsis and its interaction with thematic fit. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers*, pages 3340–3353. Association for Computational Linguistics.

Jeroen van Craenenbroeck and Tanja Temmerman. 2018. *The Oxford Handbook of Ellipsis*. Oxford University Press.