Computing Ellipsis Constructions: Comparing Classical NLP and LLM Approaches

Damir Cavar
Indiana University
dcavar@iu.edu

Zoran Tiganj
Indiana University
ztiganj@iu.edu

Ludovic Veta Mompelat
University of Miami
lvm861@miami.edu

Billy Dickson
Indiana University
dicksonb@iu.edu

Abstract

Although ellipsis constructions are highly frequent in common genres and discourse in all languages, State-of-the-art (SOTA) Natural Language Processing (NLP) technologies face significant challenges with such constructions. While the phenomenon as such is theoretically well-documented and understood, current technologies fail to provide adequate syntactic and semantic analyses due to many factors. One of those factors is insufficient cross-linguistic language resources covering ellipsis and ultimately serving the engineering of NLP solutions that more adequately provide correct analyses for ellipsis constructions. This article describes our effort to create a dataset that currently covers more than eighteen 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 demonstrate 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-linguistically reconstructing ellipsis and parsing it with SOTA NLP technologies results in acceptable representations for downstream tasks.

1 Introduction

As discussed in more detail from a typological perspective in (Cavar et al., 2024), 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.

While in discourse situations, different elements of utterances or sentences can be elided if they could be derived from the previous context, the constructions that we are interested in are ellipses in sentences without obligatory extra-sentential licensing conditions. A common ellipsis type that is licensed within sentence boundaries is forward or backward conjunct reduction, as in example (1). It is common cross-linguistically. In the examples (1), the Croatian or German counterpart of *my sister* has been elided in the underlined position.

(1) a. *Moja sestra* živi u Londonu i ___ radi u Amsterdamu. (Croatian)

b. *Meine Schwester* lebt in London und ___ arbeitet in Amsterdam. (German)

c. *My sister* lives in London and ___ works in Amsterdam.

d. *My sister* lives in London and *my sister / she* works in Amsterdam.

The possibility of eliding phrases or words in coordinated constructions has universal and language-specific aspects. Certain ellipsis constructions are common in all languages we are aware of. Depending on underlying word order constraints, whether a language is an SVO or an SOV language results in language-specific ellipsis constraints. In addition, differences in morphology and general morphosyntactic properties can lead to peculiarities in the context of ellipsis.

Ellipsis constructions like FCR are possible in all languages we are aware of. In fact, whenever possible, ellipsis is the preferred form of presentation in text or spoken language in various construction types, e.g., in coordination constructions. This means that ellipsis is applied in unmarked cases whenever it is possible. We could hypothesize that ellipsis optimizes the signal entropy and improves communication by reducing time and effort. Whenever elements that could be elided remain overt in sentences or utterances, they might indicate specific semantic or pragmatic reasons. A sentence like (1d) appears to be emphatic if the phrase *my sister* is used. Such explicit repetitions of content stand in contrast to the unmarked default in ellipsis construction (1a).

Proceedings of the Society for Computation in Linguistics (SCiL) 2024, pages 217-226.
Irvine, California, June 27-29, 2024

In *gapping* constructions, as in (2a), we see that the verb complex *was watching* is elided. In example (2b), a case of VP-Ellipsis, the entire predicate or Verb Phrase (VP) *read War and Peace* is elided.

(2) a. Paul and John were watching the news, and Mary ___ a movie.
b. Susan read War and Peace but Mary did not ___

Ellipsis constructions like gapping do not require a licensing discourse context, i.e., no context outside of the sentence boundaries is necessary to license such ellipsis. Therefore, the licensing context is purely intra-sentential.

Discourse licensed ellipsis constructions are context-dependent and extra-sentential forms of ellipsis in responses to questions, as in example (3). The words *each candidate will talk* that are spelled out in the question (3a) are elided in the response (3b) (Cavar et al., 2024).

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

There are many more very specific ellipsis types that we cannot discuss in detail in this context. Each type of ellipsis comes with specific construction properties and limitations. One additional aspect of ellipsis worth mentioning here is that the elided content does not have to match the intra-sentential licensing context.

We can find some examples in English with lexical mismatches of elided word forms and licensing context, as in 4a. In the Croatian example (4b), a highly inflecting language, the licensing context morpho-syntactically and phonological does not match the elided forms. The elided content does not have to be homophonous with the intra-sentential licensing context. In the examples (4) the round brackets indicate the elided content and contain the morpho-syntactically correct forms that could fill the gaps.

(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

A particularly problematic type of ellipsis is a scattered ellipsis of multiple words elided in different positions in a clause. In example (5) the words *will*, *greet*, and *first* are elided in the second conjunct.

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

As (Cavar et al., 2024) emphasized, and as pointed out in Testa et al. (2023) and Hardt (2023), common text genres exhibit a large number of all ellipsis types. Surprisingly, human processing is not at all impacted by ellipsis. On the contrary, ellipsis seems to improve the discourse and readability of text. The challenge for NLP processing such constructions is discussed below.

1.1 NLP Problems

One of the main issues why we experiment with ellipsis constructions is related to generating syntactic representations for subsequent semantic processing. In order to derive semantic representations and properties of utterances and sentences, we utilize functional relation annotation of sentence elements, for example, the automatic labeling of *subjects* and *objects*, or scope relations of quantifiers and operators. Common Dependency Grammar, Phrase Structure, or Lexical-functional Grammar parsers fail to analyze ellipsis constructions adequately. However, parsing ellipsis constructions with the elided elements undone or reconstructed results in significantly more useful parse trees. The examples (2a) and (2b) are not correctly parsed by common SOTA NLP-pipelines, while the examples (6a) and (6b) with ellipsis undone result in useful and acceptable parse trees.

(6) a. Paul and John were watching the news, and Mary was watching a movie.
b. Susan read War and Peace, but Mary did not read War and Peace.

Compare the parse trees generated by spaCy 3.7 using the English transformer model for (2a) and (6a) with the corresponding parse trees in Figures 1 and 2 respectively.

The problem with the DPT in Figure 1 is that the predicate head *watching* is coordinated with the direct object head in the second conjunct *movie*. At the same time, the subject in the second conjunct *Mary* is analyzed to be the subject of the direct object *movie*. With the ellipsis undone in the DPT in Figure 2, the dependency relations are correctly analyzed, resulting in a useful parse tree. These types of errors are systematic and can be replicated

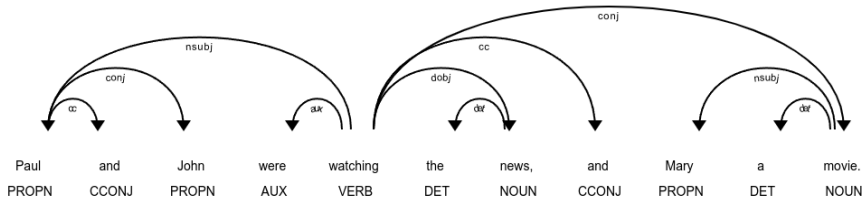

Figure 1: spaCy Dependency Tree (DPT) for example (2a).

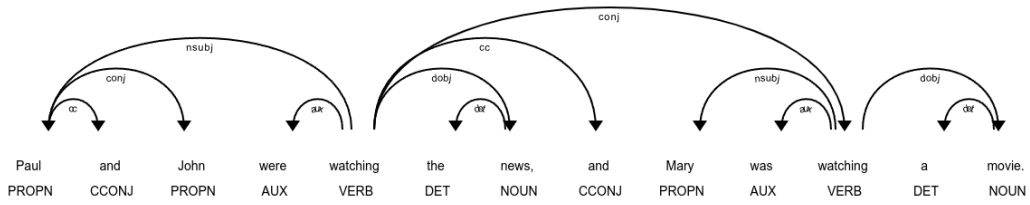

Figure 2: spaCy Dependency Tree (DPT) for example (6a).

for all our examples in The Hoosier Ellipsis Corpus (THEC) using various SOTA NLP pipelines.

Our experiments with parsing ellipsis constructions and comparing the output with ellipsis constructions undone were performed on the most recent versions of:

- Berkley Neural Parser (Benepar), (Kitaev and Klein, 2018; Kitaev et al., 2019), version 0.2.0, https://github.com/nikitakit/self-attentive-parser
- spaCy, (Honnibal and Johnson, 2015), version 3.7, https://spacy.io
- Stanza, (Qi et al., 2020), version 1.8.2, https://stanfordnlp.github.io/stanza/
- Xerox Linguistic Environment (XLE), (Crouch et al., 2011), https://clarino.uib.no/iness/xle-web

We experimented with all the languages in THEC for which we could identify models or grammar in the listed NLP pipelines. For almost all examples, the NLP pipelines generated inappropriate DPTs, Phrase Structure Trees (PST), or LFG style c- and f-structure pairs. Some of the error types are explained below.

In our evaluation of the NLP-pipeline output, the resulting trees were judged by a team of syntacticians familiar with all three relevant grammar frameworks, i.e., Dependency Grammar (DG), Phrase Structure Grammar (PSG), and Lexical-functional Grammar (LFG). Using spaCy, we were able to experiment with Chinese, Croatian, English, German, Japanese, Korean, Norwegian, Polish, Russian, Spanish, and Swedish data. With XLE, we could only use the English, Norwegian, German, and Polish grammar. Stanza does not offer PST output for most of the languages we were targeting.

In the next section, we will describe how LLMs were as challenged with ellipsis constructions as these rule-based, statistical, or neural syntactic parsers.

While in most of the cases, the Dependency parser output improves with constructions without ellipsis, errors still remain problematic. Figure 3 shows an example with gapping of the verb in the second conjunct. The parser obviously confuses the coordination relation suggesting that the subject in the first conjunct *people* is coordinated with the auxiliary in the second conjunct *do*. However, the analysis of the first conjunct is already wrong since the predicate *like broccoli* is analyzed as a nominal modifier.

The error in the counterexample without ellipsis does not improve the parse tree in Figure 3. As the parse tree in Figure 4 shows, the conjunction relation is still wrong, suggesting that *people* and *like* is conjoined. The error in the first conjunct remains the same, while now the second conjunct structure without gapping results in an acceptable representation.

It is clear from a theoretical perspective that Dependency parsers will have issues with implicit lexical material in sentences. DG is primarily concerned with dependencies between overt lexical

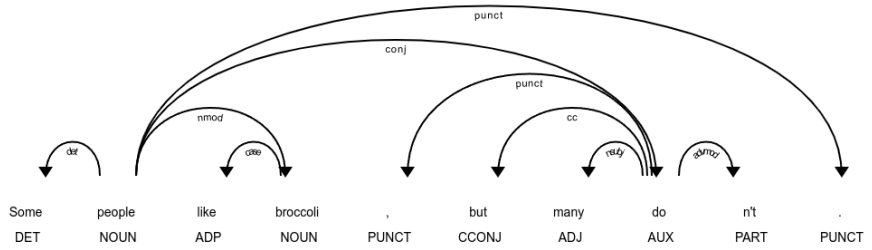

Figure 3: Stanza Dependency Tree Ellipsis 1

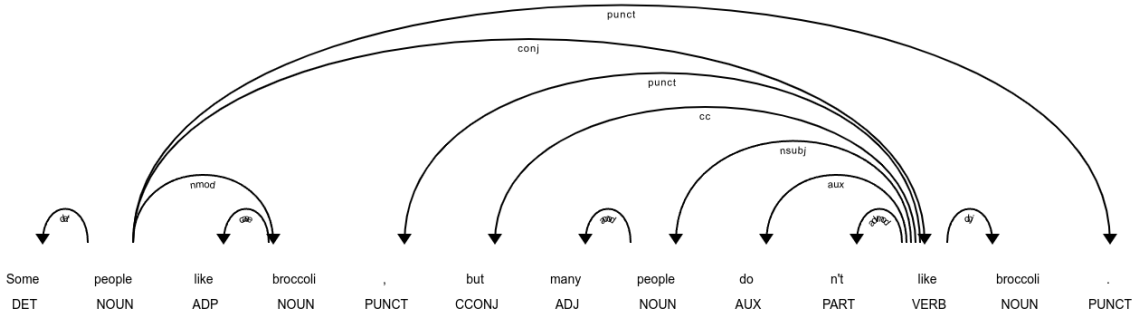

Figure 4: Stanza Dependency Tree Ellipsis 2

items and not implicit words or content.

The Stanza constituency parser does not provide a better result, as in Figure 5. It assumes the predicate head *like* to be the preposition head of a phrase modifying the subject phrase *some people*. The structure of the subordinate clause containing VP-ellipsis is useless for any further semantic post-processing.

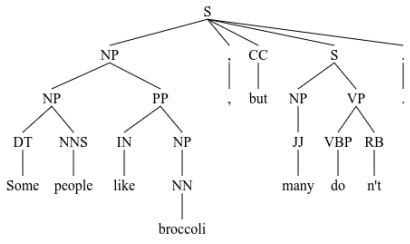

Figure 5: Stanza Constituency Tree

In our experiments, we can confirm that these examples are not rare parser errors. These are systematic mistakes that these parsers make in ellipsis constructions. The vast majority of ellipsis constructions will not be parsed correctly by current SOTA NLP pipelines, independent of the theoretical framework of the grammars or treebanks used for parser engineering, and independent of the parser model (e.g., rule-based, neural, LLM-based).

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.

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.

As mentioned above, intra-sentential licensing of ellipsis in gapping constructions is not necessarily dependent on the discourse context. Example 7 shows that complex gapping constructions are not restricted by syntactic phrase boundaries or structures, but maybe phonological conditions. None of

this is a parsing challenge for human listeners, but it is a significant problem for NLP pipelines.

(7) Jimmy was always dreaming about going to Paris, and Mary ___ to Tokyo.

Our central goal in The Hoosier Ellipsis Corpus Project is to create corpora and language resources for the evaluation and development of NLP pipelines that can generate semantically more adequate syntactic structures for ellipsis constructions.

1.2 Previous Work

To present an overview of the theoretical work on ellipsis constructions in this context is impossible. Given the vast amount of publications on ellipsis using numerous descriptive and theoretical frameworks, we encourage the interested reader to consult excellent handbooks on that topic, for example, the Handbook of Ellipsis (van Craenenbroeck and Temmerman, 2018). The following summary focuses on recent computational and corpus approaches to ellipsis constructions.

Liu et al. (2016) investigated Verb Phrase Ellipsis (VPE) and conducted three tasks on two datasets. The first dataset is the Wall Street Journal (WSJ) section of the Penn Treenbank with VPE annotation (Bos and Spenader, 2011), and the second dataset is 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. 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 *sponsor* string to fill and replace the elliptical gap. 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.

Droganova et al. (2018a,b) first created artificial treebanks containing elliptical constructions for English, Czech, and Finnish, using the Universal Dependencies (UD) (Nivre et al., 2016) annotation standard and 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.

NoEl (An Annotated Corpus for Noun Ellipsis in English) was motivated by the assumption that noun ellipsis is more frequent in conversational settings. It is described in Khullar et al. (2020), 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.

The Santa Cruz sluicing dataset is documented in Anand et al. (2021). In it, 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.

The *ELLie* corpus and related experiments are discussed in Testa et al. (2023). It is a dataset of elliptical constructions that has been evaluated using GPT-2 (Radford et al., 2019) and BERT (Devlin et al., 2019a), two Transformer-based language models, on their ability to retrieve the omitted verb in elliptical constructions that demonstrate different levels of 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.

2 The Hoosier Ellipsis Corpus

The Hoosier Ellipsis Corpus (THEC) V 1.0 (Cavar et al., 2024) consists of data from eighteen languages. It includes data from low-resourced languages like Navajo and Kumaoni. To our knowledge, this is the only collection of ellipsis examples in some of these low-resourced languages. The THEC also contains unique collections of ellipsis constructions from common Slavic languages (Russian, Ukrainian, Polish).

The corpus includes the various ellipsis types, e.g., VP-ellipsis, Sluicing, Gapping, Stripping, Forward (FCR), and Backward Coordinate Reduction (BCR). Where necessary, the previous and following context of the ellipsis is provided as well.

While continuously extended with more data and other languages, Table 1 lists the languages, and the current example counts in the THEC.

THEC data consists of sentence pairs. The Close test (Taylor, 1953) and the masked word machine learning approach taken in BERT (Devlin et al., 2019b) inspired the design of the data format. The example with ellipsis is provided, and the position of the elided content is marked with three underscores, as in Figure 6. The fully spelled-out form of the corresponding ellipsis construction is separated by four dashes in a new line, providing the elided content.

Arabic	375	Croatian	6
English	267	German	79
Gujarati	9	Hindi	127
Japanese	105	Korean	40
Kumaoni	85	Mandarin Chinese	40
Navajo	9	Norwegian	55
Polish	139	Russian	202
Spanish	171	Swedish	20
Telugu	20	Ukrainian	158

Table 1: Corpus languages and example counts

Additionally, the example entry can be accompanied by the previous or following context. The previous context is indicated by the `B:` tag (for *before*), and the following context is indicated by the `A:` tag (for *after*). In a specific comment or meta information section of lines introduced by a hashmark, as in Figure 6, the source of the example, the annotator, and a translation into different languages can be provided.

```
A Nina ___ na pianinie.
----
A Nina gra na pianinie.
B: Kasia gra na klarnecie.
A: Marek śpiewa.
# source: Marjorie J. McShan (2000)
# TR eng: Nina plays piano.
```

Figure 6: Polish THEC gapping example with additional information.

This simple Unicode text-based format for encoding allows us to focus on common machine-learning approaches for experiments using various NLP technologies. This format also allows us to annotate ellipsis constructions that contain numerous elided slots (e.g., scattered ellipsis).

We focus in this data annotation approach on indicating the distributional properties of elided content in sentences, be it discourse licensed or purely syntactic ellipsis. The goal is to reflect the 'understood' or 'implied' sequence of words as understood by human native speakers, independent of any particular syntactic theory of ellipsis.

Our goal is to convert most of the ellipsis and full-form pairs into the Universal Dependencies 2 format with correctly encoded ellipsis.[1] In this simple data format, we can add PSG-style annotations

[1]See for details the documentation for UD 2 at https://universaldependencies.org/u/overview/specific-syntax.html.

to the meta-section for every example, providing the phrase structure tree and additional syntactic information, or triple sets for dependencies, as well as c- and f-structure strings based on the LFG formalism.

The data source for the THEC is mainly literature and curated language corpora and data collections. We used mostly examples from peer-reviewed, theoretical, or documentary linguistic publications. In some cases we provide unique data that has not been published previously. In these cases the data was generated by native speakers (e.g., Navajo) and validated with their speaker communities.

3 NLP Experiments: Methods & Results

We reported in (Cavar et al., 2024) about the motivation for THEC and the first initial experiments testing NLP capabilities with the THEC constructions. Here, we expand these experiments to include new SOTA models and experimental strategies.

With the goal in mind to develop NLP pipelines that are capable of processing ellipses constructions and generating adequate representations, we defined three main tasks to test the capabilities of current SOTA NLP technologies and identify possible solutions for reconstructing fully spelled-out sentences from ellipsis constructions. The tasks involve a.) a binary classifier for the detection of ellipsis in sentences, b.) a model for the identification of the positions of elided content in sentences with ellipsis, and c.) a model for the prediction of the elided content in the correct positions in ellipsis constructions. The tasks a.) and b.) presuppose that the models are given only sentences that contain ellipses.

In (Cavar et al., 2024), we show that three different NLP approaches perform very differently and that LLMs were outperformed on task a.) by even a simple Logistic Regression classifier. The best-performing model for task a.) and task b.) was a BERT-based, Transformer-based classifier and labeler. For task c.), we could only utilize Large Language Models, of which only GPT-4 provided acceptable results for English, Spanish, and Arabic. In these initial experiments, we assumed 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 and that it is useless in a task like c.), e.g., generating the morpho-syntactically correct word forms for the elided content.

Initially, we expected transformer-based models to perform well as classifiers, we also expected them to be less efficient at guessing the position of elided content. Our expectation was also that current SOTA LLMs would be outperforming all other models in all three tasks. For generating the correct surface form of the elided content we did not see any other model beating SOTA LLMs 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), adding some corrections and modifications to it since some native speakers complained about the naturalness of some of the ellipsis constructions. We also used some 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, Ukrainian, 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 almost completely 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.

For Task 1, the classification of ellipsis, we generated tuples with the sentence and label 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 indi-

cators in the ellipsis example sentence to be able to train labeling algorithms that predict the ellipsis position or to evaluate predicted ellipsis positions directly.

3.2 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[2] 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.

For GPT-4 we used context *Classify the following sentence as containing ellipsis or not. Ellipses indicates gapping, pseudogapping, stripping, and sluicing. Answer with only 0 for sentences without ellipsis or only 1 for sentences with ellipsis.* which preceded each sentence.

We additionally conduct few-shot experiments on GPT-4 in which the model is given 4 example annotations in addition to its prompt. These examples are omitted when calculating results.

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 (7), 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.

For GPT-4 we used a prompt with a rich context: *Annotate the following sentence by placing ___ in the position of each ellipsis. Ellipses indicates gapping, pseudogapping, stripping, and sluicing. If there are no ellipses, answer with only original sentence.* We additionally conduct few-shot experiments on GPT-4. Accuracy is calculated by comparing the correctly annotated sentence to the generated GPT-4 sentence.

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 indicates gapping, pseudogapping, stripping, and sluicing. Answer with only the new sentence. If there are no ellipses, answer with only the original sentence.* We additionally conduct few-shot experiments on GPT-4.

4 Results

We tested GPT-4 (gpt-4-turbo-2024-04-09) zero-shot and few-shot, BERT, and LR. Alongside our binary LR classifier, we tested GPT-4 (gpt-4-turbo-2024-04-09) and BERT. For GPT-4, we tested on our dataset of English, Arabic, and Spanish. The results for task 1 are given in Table 2.

Model/Language	en	es	ar
LR	0.74	-	-
BERT	**0.94**	-	-
GPT-4 zero-shot	0.59	0.73	0.61
GPT-4 few-shot	0.64	**0.75**	**0.73**

Table 2: Task 1 Binary Classification Accuracy for English, Spanish, and Arabic

It is surprising that the GPT-4 zero-shot classi-

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

fication is worse than the LR-baseline, and significantly worse than the BERT-based classifier. Precise scores from the zero-shot and few-shot GPT-4 experiments are given in Table 3.

GPT-4 zero-shot			
Language	**f1**	**p**	**r**
English	0.63	0.54	0.76
Spanish	**0.70**	**0.60**	**0.85**
Arabic	0.33	0.33	0.33
GPT-4 zero-shot			
Language	**f1**	**p**	**r**
English	0.65	0.60	0.70
Spanish	**0.70**	**0.63**	0.79
Arabic	0.67	0.52	**0.94**

Table 3: Precision, Recall, and F1-Score for GPT-4 across English, Spanish, and Arabic

Given the default temperature setting of 0.7 in GPT-4, the output from the model is not deterministic for a given input sentence. In order to reduce randomness in the model, we set the temperature of GPT-4 to 0. This approximates the model choosing a response that it deems most probable, instead of it sampling from possible responses.

In Task 2, we tested an initial BERT-based ellipsis position guesser and GPT-4 zero-shot and few-shot. Task 2 results are shown in Table 4.

model/language	**en**	**es**	**ar**
BERT	**0.70**	-	-
GPT-4 zero-shot	0.18	0.27	0.07
GPT-4 few-shot	0.26	**0.34**	**0.15**

Table 4: Task 2 Ellipses Location Identification Accuracy for English, Spanish, and Arabic

Surprisingly, BERT achieved an accuracy of 0.7. The GPT-4-based experiments on Task 2 were challenging. The prompt engineering for the zero-shot experiment resulted in a low accuracy across all languages. For Task 3, we exclusively focused on the evaluation of GPT-4. Results for Task 3 are shown in Table 5.

model/language	**en**	**es**	**ar**
GPT-4 zero-shot	0.22	0.29	0.01
GPT-4 few-shot	**0.34**	**0.42**	**0.35**

Table 5: Task 3 Elided Word Generation Accuracy for English, Spanish, and Arabic

GPT-4 performed better on elided word generation than ellipses location identification, however this remained a difficult task with low accuracy across all languages. In all tasks, few-shot improved GPT-4 performance.

5 Conclusion

Ellipsis constructions are obviously still challenging for all the common SOTA NLP pipelines and rule-based systems. 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 for the description of 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.

The Ellipsis Corpus and the relevant code for the experiments described in the article are available 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.

Damir Cavar, Ludovic Mompelat, and Muhammad Abdo. 2024. The typology of ellipsis: A corpus for linguistic analysis and machine learning applications. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 46–54, St. Julian's, Malta. Association for Computational Linguistics.

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. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

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.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder.

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 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.

Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

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.