Conjunct Resolution in the Face of Verbal Omissions

Royi Rassin[1] **Yoav Goldberg**[1,2] **Reut Tsarfaty**[1]
[1]Bar-Ilan University [2]Allen Institute for Artificial Intelligence
{rassinroyi, yoav.goldberg, reut.tsarfaty} @gmail.com

Abstract

Verbal omissions are complex syntactic phenomena in VP coordination structures. They occur when verbs and (some of) their arguments are omitted from subsequent clauses after being explicitly stated in an initial clause. Recovering these omitted elements is necessary for accurate interpretation of the sentence, and while humans easily and intuitively fill in the missing information, state-of-the-art models continue to struggle with this task. Previous work is limited to small-scale datasets, synthetic data creation methods, and to resolution methods in the dependency-graph level. In this work we propose a *conjunct resolution* task that operates directly on the text and makes use of a *split-and-rephrase* paradigm in order to recover the missing elements in the coordination structure. To this end, we first formulate a pragmatic framework of verbal omissions which describes the different types of omissions, and develop an automatic scalable collection method. Based on this method, we curate a large dataset, containing over 10K examples of naturally-occurring verbal omissions with crowd-sourced annotations of the resolved conjuncts. We train various neural baselines for this task, and show that while our best method obtains decent performance, it leaves ample space for improvement. We propose our dataset, metrics and models as a starting point for future research on this topic.

1 Introduction

Natural language is economic, and many elements in the message are not spelled out but omitted by speakers, left for the receiver of the message to complete. The hearer, either a human or an algorithm, should then complete the missing information and recover the intended meaning. This kind of omission and recovery occurs at all levels of conversation, from syntax to pragmatics, and is performed naturally and intuitively by humans, to the extent that they often don't even realize that something was missing in the message they received.

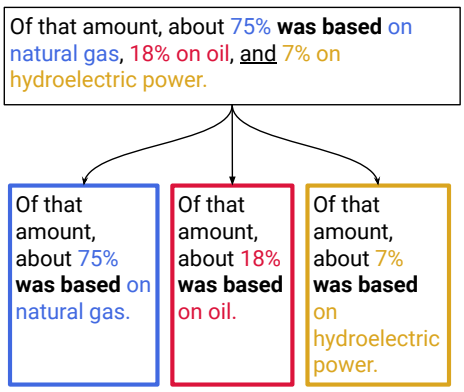


Figure 1: A demonstration of the conjunct resolution task. At the top, the input sentence with the omitted words in bold and the coordinating element is underlined. Below, the expected rewrite, consisting of a single fully-specified standalone statement for each of the conjuncts. The rewrite must not include the coordinating element.

An important class of omission phenomena — and the focus of this work — involves verbs and their arguments, especially around coordination structures. A verb and some of its arguments that appear in an initial clause may be omitted in subsequent clauses. For example, in the sentence "Josh likes wine and Jane water", the verb *likes* is omitted from the phrase "Jane likes water" and has to be inferred. We find that state-of-the-art syntactic parsers (Honnibal and Montani, 2017; Nguyen et al., 2021) fail consistently even on toy examples such as this one, and assign a structure in which "Jane water" is interpreted as a noun-compound which is the object of Josh's liking, together with water (Figure 2). Downstream applications, such as, Google translate[1] also fail. Google translate translates the above sentence to French as "Josh

[1]https://translate.google.com/, accessed on Jan 18 2023.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13623–13640

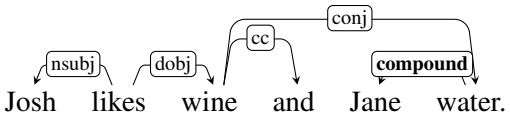


Figure 2: Compound relation indicating a misinterpretation of the sentence "Josh likes wine and Jane water" instead of "Josh likes wine and Jane likes water"

aime le vin et l'eau de Jane". When back-translated to English using the same engine, it results in "Josh likes Jane's wine and water". Significantly more complex sentences involving verbal omissions are of course also possible, and NLP systems routinely fail around them.

Linguistically, such verbal omissions are studied under the terms *ellipsis* and *gapping*, and the research broadly aims at categorizing the verbal omission cases into sub-categories, carefully documenting the condition under which different omissions can or cannot occur (Hudson, 1989; Ross, 2014; Jackendoff, 1971). For language technology purposes, however, we would like to focus on detection, resolution and usability, not categorization. The need to recover empty elements automatically and robustly in large-scale calls for a unifying approach, by which to consider the different cases of verbal omissions as instances of a single phenomenon, that can be effectively addressed by a broad-coverage NLP system.

Due to failures of even language technology applications on such constructions, in this paper, we suggest that these verbal omission phenomena should be addressed explicitly by systems, rather than hoping it will be caught on as a side product of end-to-end neural training. To this end, our aim is to establish a verbal-omission recovery task, and to create a supporting large-scale corpus documenting such verbal omissions in naturally occurring English sentences, together with automatic annotation models that recover the implicit information and make it explicit.

How should a system that resolves such missing information look like? What are its outputs? In the syntactic parsing and treebanking literature, there has been an ongoing debate regarding the best way to represent recovery of omissions in coordinated structures (Marcus et al., 1999; Nivre et al., 2020; Schuster et al., 2017; Hudson, 1973; Nielsen, 2004; Anand and Hardt, 2016; Park and Kang, 2007; Park, 2009; Ficler and Goldberg, 2016; Kato and Matsubara, 2020; Droganova et al., 2018b,a).

However, none of the solutions are fully satisfactory. First, all of them are highly technical in nature, and require significant linguistic expertise in order to even understand the notation. Often they assume some formal tree or graph notation, which makes their adoption unlikely by people who work on NLP systems but who may lack the specific syntactic expertise. Moreover, how can one feed such theory-loaded graph-based annotation into NLP systems for downstream, user-facing tasks?

To mitigate this, our proposed solution is strictly within the text-to-text paradigm, and involves a rewriting task where the input is a sentence exhibiting a coordinated structure, and the output is the set of sentences which contain the same information, but where the implicit information is made explicit. We illustrate an input and output example in Figure 1. As we further discuss below, this text-enrichment approach has several appealing properties: first, it is natural and easily comprehensible to any competent speaker of the language. Second, this caters for both large-scale annotation efforts and task adoption by potential users. Lastly, it produces an output which can trivially be fed into any language processing system which takes natural language text as input.

We collect a corpus of 10,206 sentences involving a wide range of verbal omissions in coordinated structures, annotated via this text-rewriting task to recover the missing elements. Using this corpus, we conduct experiments on the omission-recovery task. Training a T5-based model to perform the task shows that the performance saturates after seeing only 10% of the data, but the accuracy of the neural model is far from being perfect, leaving ample room for future modeling and improvements.[2]

2 Related Work

Previous research on verbal omissions in coordination structures has classified such phenomena into (at least) six categories: (1) conjunction reduction (Hudson, 1973); (2) gapping (Ross, 2014; Jackendoff, 1971; Hudson, 1989); (3) VP Ellipsis (Nielsen, 2004); (4) sluicing (Anand and Hardt, 2016; Park and Kang, 2007); (5) pseudo-gapping (Park, 2009); and (6) argument clusters. All of these phenomena are forms of ellipsis and are considered as ways to use language efficiently.

How do NLP systems cope with such complex

[2]We make our code and dataset publicly available at https://github.com/RoyiRa/conjunct-resolution-task

syntactic phenomena? Work around gapping in the syntactic parsing community has focused on representation schemes for this phenomena in dependency graphs. Various representation has been proposed. For instance, the Universal Dependencies (UD) framework (Nivre et al., 2020) introduces the concept of an "orphan" dependency to indicate the presence of an ellipsis, and (Schuster et al., 2017) provides a detailed analysis of how gapping constructions could be represented using UD. Related work (Schuster et al., 2018; Droganova et al., 2018a,b) utilize such schemes by promoting a new head of the clause in cases of gapping and attaching all remnants to it.

However, having a representation schema for a construction does not mean that an automatic parser is able to accurately predict it. To bridge this gap (no pun intended), several efforts have been made to increase the training data size for these constructions by enriching existing data or creating artificial datasets. Droganova et al. (2018b) proposed data enrichment methods that utilize existing annotated parse trees to mimic the structure of elliptical constructions, and Droganova et al. (2018a) trained a parser on artificial elliptical treebanks, achieving an F1 score of 36% on a small dataset (an improvement relative to prior work). Moreover, Schuster et al. (2018); Kato and Matsubara (2020) proposed a reconstruction algorithm at the dependency graph level, but relied on an oracle to identify the gapped sentences.

Despite these efforts, current approaches do not fully address the issue of verbal omissions in coordination structures, and remain scattered. Additionally, these methods are not suitable for large-scale applications and are not ready for use in downstream tasks. In this work we take a more realistic, consistent and unified approach, wherein all verbal omissions are treated within the same framework, for which we provide a gold benchmark, a crowdsourcing interface, models for the tasks demonstrating feasibility and efficacy.

3 The Conjunct Resolution Task

3.1 Desiderata

We seek a unifying approach to handle many types of verb-related omissions in coordination structures, and which is targeted primarily at users of language technologies. Concretely, our approach should allow for (1) an annotation scheme that is scalable to multiple annotators while maintaining quality; (2) a comprehensible task to non-linguists that is simple enough that no specialized linguistic expertise is required by a user of the resulting annotations; (3) amenable to automatic annotation by models; (4) useful in the context of a downstream language processing applications. Additionally, it would be preferable if the approach is language-agnostic, and that the approach is not constrained by any particular linguistic theory.

3.2 The Task

In order to meet the aforementioned desiderata (Section 3.1), our approach steps away from traditional syntactic representation and instead represents the output as natural language text, which is an enriched version of the input text. This offers several advantages. First, the resulting annotation task is intuitive for annotators as they now need to read a sentence and rewrite it, which is a natural and familiar setting; second, language models are designed to learn from, operate on, and produce natural text representation; finally, the output can be consumed by any process that takes text as input, so applications can benefit from the enrichment without requiring a change in their design.

Intuitively, such a task could be to take a sentence with missing information, and rewrite it by completing in all the missing verbs and their arguments, e.g. rewriting “Josh likes wine and Jane water” to “Josh likes wine and Jane likes water”. However, we found this task to be notoriously hard to explain both to non-linguists (who did not realize a verb was missing in this sentence in the first place) and to linguists (who also do not identify some of the verbs as “missing”, for example in “Jane likes wine and water”, which is not technically a gapping construction but a verb taking a coordinated NP as its object — but which we aim to reconstruct as two distinct conjuncts[3] nonetheless).

Instead, we build on the *split-and-rephrase* paradigm (Narayan et al., 2017), which involves breaking down a complex sentence into smaller, simpler sentences. Specifically, we propose to decompose sentences that potentially involve verbal omissions around coordination into a set of independent sentences, that together capture the meaning of the original sentence, and do not add to it. In contrast to the original split-and-rephrase work, where no information is actually missing, the

[3] In this work we use the word “conjuncts” to mean expressions linked by the conjunctions “and”, “or”, or “but”.

rewritten sentences make the implicit arguments explicit in each conjunct, and when they are taken together, these sentences retain the meaning of the original complex sentence.

Concretely, we define the conjunct resolution task as follows: given a sentence containing one of the conjunctions “or", “and", or “but" as input, the sentence has to be rewritten into a set of sentences while adhering to the following constraints:

> (1) the set of sentences must not include the marked conjunction;
> (2) the sentences should introduce a minimal number of new content words;
> (3) the sentences set should preserve the meaning of the original sentence and not add to it;

If it is not possible to rewrite the sentence under the preceding constraints without altering its meaning, the sentence should be left unchanged.

The first constraint drives the annotation: by not allowing to use the conjunction, the sentence must be split, and all the verbs and their arguments must be spelled out. The two other constraints keep the resulting sentence set both minimal and complete.

To illustrate our task, consider the following sentence. The underlines indicate omitted elements, and are not part of the input. The focused conjunction "and" is marked in bold:

- “*As of January 2013, The Times has a circulation of 399,339, The Sunday Times __ of 885,612,* ***and*** *The New York Times __ of 9,512,132.*”

The sentence is rewritten into a set of three sentences as each clause describes a unique event:

(1) As of January 2013, The Times has a circulation of 399,339.

(2) As of January 2013, The Sunday Times has a circulation of 885,612.

(3) As of January 2013, The New York Times has a circulation of 9,512,132.

A core challenge of this task is to rewrite the sentence to a correct number of sentences while faithfully retaining the meaning of each clause (for instance, including the opening span "As of January 2013," is crucial in retaining the overall meaning of the sentence, however, it should not be a sentence on its own, as it is not a coordinated clause).

The closing clause refers to sentences that can not be rewritten to independent sentences, but still contain verbal omissions. For instance, consider

- “*Amla made 133* ***and*** *Roussow __ 132 with the pair combining to put on 247 for the third wicket.*”

while this is indeed a case of verbal omissions, the span “*with the pair combining to put on 247*” binds the two clauses together in a way that will lose its meaning if rewritten to a set of two sentences.

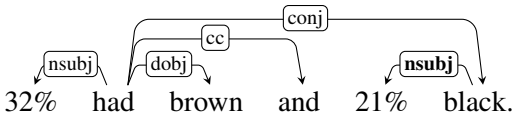
This paradigm fits our objective well and addresses the desiderata we put forth in Section 3.1, as rewriting a sentence to a set of sentences indirectly resolves the verbal omissions. It is amicable to non-linguist annotators and straightforward to scale while maintaining quality, as there is little room for variance when breaking down a complex sentence. Finally, users of the process can clearly and intuitively understand its intended behavior and can analyze its correctness, without requiring any linguistic training.

4 Data Collection Process

We collect a dataset of 10,206 examples of a wide array of omission cases, which can serve for both training and evaluation. We aim for the collected sentences to cover a wide range of omission cases. The underlying data was sourced from three publicly available datasets: SQuAD 2.0, Dailymail, and CNN (Rajpurkar et al., 2018; Hermann et al., 2015), with a roughly equal proportion of sentences from each one. Our proposed collection protocol involves two steps: (1) automatic collection of sentences that are likely to contain interesting omission phenomena; (2) manual annotation of the sentences via crowd-sourcing.

4.1 Sentences Collection

Instead of trying to explicitly target and identify specific types of verbal omissions, we instead rely on the observation that these verbal omission constructions affect both manual and automatic syntactic analysis in various ways, either due to constructions that are hard or impossible to represent, or due to parsing mistakes. We thus do not look *directly* for specific verbal omissions, but rather, identify their side-effects as manifested in the graph outputs of a dependency parser. For example, here:



the omission of *had* manifests in two "suspicious" structures: a *conj* relation between two different part-of-speech tags, and an *nsubj* dependent of a word which is not a verb. We collect cases based on 21 such patterns, applied to sentences that include coordination. By considering a sample of the sentences identified in this manner, we verified that roughly 92% of the resulting cases are indeed non-trivial verbal omission cases.

4.2 Annotation and Curation

We devise a scalable annotation procedure that can be performed by non-expert annotators.[4] The procedure is based on the conjunction resolution task (Section 3), which does not require annotators to possess advanced linguistic knowledge. Instead, it relies on their intuitive understanding of language.

Crowdsourcing Infrastructure. We set up an Amazon Mechanical Turk (AMT) task in which workers were given a coordination structure with suspected omissions and a highlighted conjunction, and were asked to rewrite the sentence into multiple independent sentences, according to our task's rules. Each AMT assignment (known as a "HIT") begins with a brief description of the task and two examples: one example of a rewritable sentence and the other of a not rewritable sentence. Workers were also given an option to view five rewritable and five non-rewritable examples with detailed explanations.

A HIT consisted of seven pairs consisting of a coordination sentence and a highlighted conjunction in it. The annotators were requested to rewrite the sentences in the order in which the clauses are read in the sentence. We encouraged annotators to review their work by joining the set of sentences with the highlighted conjunction and comparing the meaning between the input and the sum of their sentences. Moreover, when annotators submitted a sentence unchanged, they were prompted to explain why. This not only facilitated critical thinking on their part, but also allowed for potential revisions to their annotation and provided valuable insight on the data, being a useful resource on its own.

To ensure the quality of the annotations, we implemented several checks: (1) We ensured that no two sentences in the rewritten set were identical. (2) We verified that the highlighted conjunction was not present in any of the sentences in the set. (3) We confirmed that no new content words were added, while still allowing for inflectional variations in verb and noun forms to maintain grammatical accuracy. Additionally, to handle unexpected cases and gain further understanding of the task, annotators were given the option to indicate uncertainty in their annotation and to specify if the sentence was a "long list" requiring more than ten rewrites. In case of the latter, sentences were removed. Annotators were also given the option to provide any feedback. appendix A.3 shows the user interface of this task.

Annotations Consolidation. The final annotations were determined by majority agreement among annotators on the number of sentences in a set and the exact match for each submission. In cases where no majority agreement was reached, the answer provided by the highest-performing annotator was chosen.

Inter-Annotator Agreement. We assessed the level of unanimous agreement on factors such as rewrite agreement (the number of sentences required to accurately rewrite a given sentence), exact match, and average Jaccard Similarity[5]. The initial annotation phase involved 64 native English speakers with a high approval rate (99%) and significant experience on the AMT platform (over 5,000 completed HITs). Our analysis of the first 10% of the data revealed rewrite agreement, exact match, and average Jaccard Similarity scores of 56%, 67.5%, and 94%, respectively, with approximately 5% of the data being unusable due to corrupted annotations. In order to continually improve the quality of our annotations, we narrowed our pool of annotators to the top five performers (based on activity and IAA performance) and provided personal feedback and bonuses based on the execution of each batch. As a result of these efforts, unanimous rewrite agreement, exact match, and average Jaccard Similarity all increased to 85%, 82%, and 97%, respectively, and less than 1% of the data required corrections.

[4] We rely on a pool of trained crowd workers in the controlled crowd-sourcing paradigm (Roit et al., 2019).

[5] Jaccard Similarity is defined as the size of the intersection of the sets divided by the size of the union of the sets.

5 Conjunct Resolution Dataset

Our Conjunct Resolution dataset consists of 10,206 verbal omission sentences, each paired with one of the conjunctions: "and", "or", and "but" (table 2 reports conjunction distribution) coupled with human annotations. By subtracting the number of verbs in the verbal omissions to those in the gold annotations, we find that 42% of the verbs are omitted (see table 1). Furthermore, the majority (95.2%) of sentences were found to be rewritable, with 82% of the sentences being expressed in two sentences, 9.8% being expressed in three sentences, 3.4% expressed as four sentences or more, and only 4.8% being classified as not rewritable.

Split	Explicit	Omitted	Total
Train	29,447	21,355	50,802
Validation	3,630	2,631	6,261
Test	3,611	2,517	6,128
Full	36,688	26,502	63,190

Table 1: Count of explicit and omitted verbs in each split of the dataset, and the total count for each split. The full dataset contains a total of 10,206 instances

Split	and	or	but	Total
Train	6,508	798	860	8,166
Validation	805	108	108	1,021
Test	811	90	118	1,019
Full	8,124	996	1,086	10,206

Table 2: Distribution of conjunctions (and, or, but) in each split of the dataset, and the total count for each split. The full dataset contains a total of 10,206 instances

Non-rewritable Sentences. 491 of the 10,206 (4.9%) were marked as non-rewritable. Of these, 445 contain an explanation.[6] The reasoning behind deciding whether a sentence is rewritable or not seems to be non-trivial. For instance, consider the following two sentence:

(1) I'd say Adam **will win** four majors and Justin __ three, but I wouldn't be surprised if it was the other way round.

(2) The pair are tied at the top after McIlroy **shot** 67 – his 25th score under par out of his last 27 rounds - and Horschel __ a 69.

Despite that both sentences are cases of gapping, only the second is rewritable. To recognize this, the reader (human or model) needs to be able to identify when two events are bound together by another piece of information. In the first sentence, the phrase "but I wouldn't be surprised if it was the other way round" lacks context when appearing in the rewritten sentences. However, for the second sentence, there is no such issue, and is thus rewritable.[7]

6 Evaluation Metric

As no task is complete without an evaluation metric, we propose an automatic evaluation metric for the proposed conjunct resolution as rephrasing task. For evaluation, we are interested in measuring three things: (1) how accurate the model is at resolving verbal omissions, (2) how often does the model omit other information after resolution, and (3) how often the model generates extra information. To address these three criteria, we propose to measure recall and precision over the *predicate-argument relations* recovered by a dependency parser on the generated compared to the gold sentence set.

High Level Description. The task revolves around making verbs and their arguments explicit. Our main object of interest is thus the "verb nucleus", an instance comprising of a verb and its arguments, as reflected in the dependency tree. We measure to what extent the nuclei extracted from the generated sentences overlap with the nuclei extracted from the gold sentences. Neglecting to resolve an argument, or adding an extra argument to a given verb, will result in a mismatch between the gold and generated nucleus of that verb, hurting both recall and precision. Neglecting to spell out a verb completely will result in a missing nucleus (recall error), and over generating will result in spurious nuclei (precision error).

To calculate these metrics, we first produce dependency graphs for both the model's generated set of sentences and the gold annotation's set of sentences. From these graphs, we extract the verbs, and for each verb a subset of its dependents that we consider as arguments (based on a set of dependency labels). Each such set of verb+argument is

[6] We did not collect explanations during the first few batches.

[7] In linguistics and formal semantics, when a coordinated structure refers to the plurality of events as a whole, it is said to have a *collective* (as opposed to *distributive*) reading. The non-rewritable sentences in our set are those with collective readings. Their annotation and resolution is beyond the scope of this paper, and we reserve them for future work.

a "verb nucleus". We treat the collection of verb nuclei over all sentences as a set (each element in the set is a collection of verb+arguments), and compute precision and recall over this set.

Details. Denote the (automatically produced) syntactic dependency graphs of the m gold sentences as $G = \{g_1, g_2, \ldots, g_m\}$, and the dependency graphs of the n generated sentences as $H = \{h_1, h_2, \ldots, h_n\}$. We extract verb nuclei from these graphs, where each nucleus is a subgraph containing a verb, its subject, object and lexicalized prepositional modifiers, as well as prepositional modifiers of the object and associated negations, if they exist.[8] We represent a nucleus as a bag of (w_1, dep, w_2) triplets where w_1 and w_2 are words and *dep* is a dependency label, and consider two nuclei to be the same if their bags are the same. Denote by N_G the bag of gold nuclei, by N_H the bag of generated nuclei, and by N_I the bag of identity nuclei, obtained by extracting nuclei from the input sentence. We obtain the subsets $N'_H = N_H \setminus N_I$ and $N'_G = N_G \setminus N_I$, which strictly contain nuclei with omitted verbs. Then $\textit{precision} = |N'_H \cap N'_G| \setminus |N'_H|$ and $\textit{recall} = |N'_H \cap N'_G| \setminus |N'_G|$. In cases where there is only one sentence in the gold set and the generated set, we skip the interaction with the identity nucleus, as to not punish the model for correctly not rewriting.

Calibration. The input sentence contains some verbs and arguments that are repeated throughout the sentence. Thus, an approach that copies the full sentence two or more times will also yield some success under our metric, due to repeated verb nuclei, despite not being meaningful. To calibrate for this, we provide two additional numbers: (a) the precision and recall obtained by a model that spits the original sentence as output, unmodified; and (b) the precision and recall of a model that has access to the correct number k of sentences in a gold annotation set, and which uses this information by spitting out k copies of the input sentence. Model performance should always be judged in comparison to these baselines.

[8]The prepositional modifiers of the object are needed to handle pp-attachment errors of the parser. We refer the reader to appendix A.5 for a comprehensive account of the parser and included arguments.

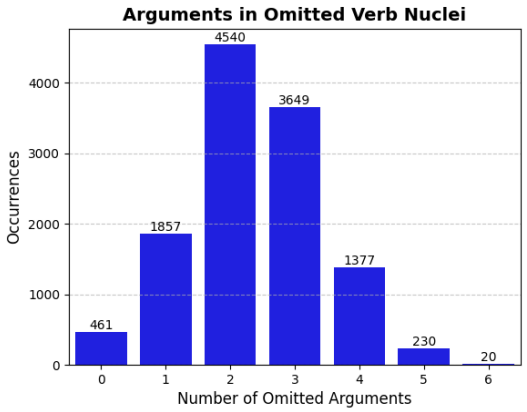


Figure 3: A distribution over the number of omitted arguments in a verb nucleus

7 Experiments

We evaluate neural models on the task, both in supervised fine-tuning and in in-context learning setups. For the supervised fine-tuning case, we measure both task performance as well as the dependence of performance on dataset size. As a concluding experiment, we take the best-performing model and manually evaluate it against the task definition.

Dataset Split and Preprocessing. We shuffle and then split our dataset to train (80%), validation (10%), and test (10%) sets. Each input instance contains a sentence and a marked conjunction. Each output instance is a sequence of sentences. For feeding the text to the models, we mark the conjunction using a special token[9] and separate the output sentences using special tokens.[10]

Supervised Fine-tuning. We fine-tune ten T5-large (Raffel et al., 2020) models, using increasingly larger subsets of the training data, from 10% to 100% in increments of ten. All models were trained with the same hyperparameters and fine-tuned for five epochs, with the best performing model on the validation set being saved and subsequently evaluated on the test-set (for the detailed training configuration, see A.2).

In-context Learning / Prompting. We evaluate the state-of-the-art GPT `text-davinci-3` model from OpenAI, in an in-context learning (prompting) fashion (Brown et al., 2020).

[9]In T5, sentinel tokens were employed, and in GPT3, the "<SPLIT>" marker was utilized.

[10]In T5, sentinel tokens were employed again, and in GPT3, each sentence was written in a separate line.

To obtain in-context examples, we randomly sampled for each test instance three re-writable and one non-rewritable sentence, sharing the conjunction with the test instance. Further details of the prompt and parameters are available in the appendix A.2.

Manual Evaluation. Our evaluation metric cannot fully evaluate the semantic correctness of results. To overcome this, we perform a manual evaluation in which a human annotator is requested to examine the system's inputs (sentences containing omissions) together with outputs of the best performing-model, and assess whether the generated set of sentences has the same meaning as the input sentence, or a different one.

7.1 Results and discussion

The results of all models are summarized in table 3. Results improve rapidly, but then saturate on around 82% F1 already with 40% ($\sim 3,200$) training samples, reaching a peak of 82.4% F1, indicating that the key to the task may not be "more data".

In our experiments, the $T5_{40\%}$ and $T5_{80\%}$ models demonstrated the best performance. However, the $T5_{80\%}$ model had a more balanced performance across the different metrics, making it the preferred model for further analysis and interpretation of results. When examining performance on specific conjunctions, the best performing model, $T5_{80\%}$, scored 83.7% on "and" sentences, 76.3% on "or" sentences, and on "but" sentences, it scored 77.7% F1. $GPT3$ performed similarly, but to a lesser extent, scoring 73.5%, 70.4%, and 65.4% F1 on average, in the aforementioned order.

In terms of quantity, at a minimum, models learned that the resolution is centered around the verb. $T5_{80\%}$ generates 5,514 out of the 6,128 (89.9%) verbs in the gold annotations, while only over-generating 349 verbs, a mere 5.7% of the total verbs in the test set. Similarly, $GPT3$ generates 4,680 out of the 6,128 (75%) and over-generates 273 verbs (4.4%).

Finally, the manual evaluation evaluating the semantic correctness of the $T5_{80\%}$ system's outputs with respect to the input, reveals that in 87.9% of the cases the meaning is preserved, and in 12.1% it is not. Although measuring different aspects of the answer, these numbers are similar to the automatic F1 results, establishing some additional trust in it as an automated metric.

Model	Recall	Precision	F1
Calibration$_1$	5.1	5.1	5.1
Calibration$_k$	49.8	41.8	45.5
$T5_{10\%}$	75.9	43.2	55.1
$T5_{20\%}$	77.2	78.2	77.7
$T5_{30\%}$	79.5	79.6	79.5
$T5_{40\%}$	**82.4**	81.8	82.1
$T5_{50\%}$	81.6	82.1	81.8
$T5_{60\%}$	81.2	81.5	81.3
$T5_{70\%}$	81	82.2	81.6
$T5_{80\%}$	82	**82.7**	**82.3**
$T5_{90\%}$	81.2	82.1	81.6
$T5_{100\%}$	81.8	**82.7**	82.2
$GPT3$	68.7	76.2	72.3

Table 3: Comparison of model performance in terms of recall, precision, and F1 score on the test set. The T5 models trained on 40% of the training set or higher generally perform similarly, with $T5_{80\%}$ achieving the highest overall F1 score of 82.3. The GPT3 model has a lower overall performance, with an F1 score of 72.3. Calibration$_1$ and Calibration$_k$ are the results of models that output back the input sentence, or k instances of the input sentence, where k is the number of sub-sentences in the rewritten text.

7.2 Error Examples

Below are some examples of errors by the best-performing T5 model.

Attributing the correct arguments to a different verb. Here, the gold annotation indicates the verb "playing" as the omission, while the model wrongly chose "gain" (crossed items are missing in output):

Players gain *points from* playing *in the four grand slams, ATP World Tour events including eight Masters Series tournaments, and World Group singles matches in the Davis Cup teams competition.*

(a) *Players gain points from playing in the four grand slams.*
(b) *Players gain points from ~~playing in~~ ATP World Tour events including eight Masters Series tournaments.*
(c) *Players gain points from ~~playing in~~ World Group singles matches in the Davis Cup teams competition.*

Incorrectly attaching two arguments to the same verb. Here the model should have produced a different sentence for "over 300,000 injured":

The January 12 quake left over 220,000 dead, over 300,000 injured, and over a million homeless.

(a) *The January 12 quake left over 220,000 dead,* ***over 300,000 injured****.*
(b) *The January 12 quake left over a million homeless.*

Failure to resolve verb-phrase ellipsis.

Germany knows how to do it, and we don't.

(a) *Germany knows how to do it.*
(b) *We don't ~~know how to do it~~.*

Considering a different coordination. Here, the model split around ';' rather than around 'and', missing information in the process:

Johnston's one shot went through her front door and over the officers' heads; they responded with 39 shots, hitting Johnston five times.

(a) *Johnston's one shot went through her front door __.*
(b) *the officers responded with 39 shots, hitting Johnston five times.*

Missing conjuncts from the resolution. Here, the crossed item did not appear in the model's output, but should have:

At the 2001 Census, 92.4% of the city's populace was White—including one percent White Irish—3.8% were South Asian, 1.0% Black, 1.3% Chinese or other ethnic groups, and 1.5% were of Mixed Race.

(a) *At the 2001 Census, 92.4 % of the city's populace was White—including one percent White Irish.*
(b) *~~At the 2001 Census, 3.8 % of the city's populace was South Asian.~~*
(c) *At the 2001 Census, 1.0 % of the city's populace was Black.*
(d) *At the 2001 Census, 1.3 % of the city's populace was Chinese or other ethnic groups.*
(e) *At the 2001 Census, 1.5 % of the city's populace was of Mixed Race.*

Resolving creates a factual inaccuracy. Here, the model splits a sentence to create factually incorrect sentences:

Heist had been sentenced to three years of probation for the identity theft and for giving false information to a law enforcement officer.

(a) *Heist had been sentenced to three years of probation for the identity theft.*
(b) *Heist had been sentenced to three years of probation for giving false information to a law enforcement officer.*

This error example illustrates an inherent limitation of our approach, as the correct semantics cannot be represented as a set of sentences. The correct behavior under our representation would have been to not split this sentence at all.

8 Conclusions

We present a novel approach for studying verbal omissions in coordination structures. Previous research in this area has been fragmented, focusing on individual phenomena. In contrast, we propose a unified approach which considers all conjunction related verbal omissions under the same framework, by introducing a text-to-text conjunct-resolution task, to resolve omitted verbs and their arguments. We compiled and curated a large dataset of conjunction related verbal omissions, consisting of over 10,000 sentences and human annotations, which serves as a valuable resource for further research in this area. Our results using state-of-the-art models as neural baselines demonstrate that this task is challenging and merits further work.

Limitations

One unsatisfying aspect of proposed task is that it accounts for *distributive* coordination structures, but is not able to handle sentences with *collective* reading where the main predicate applies to the plurality of conjuncts as a whole. In our data collection these account for about 4.9% of the verbal omission cases, and such sentences are left "non-rewritable". In future work, we would like a solution that allows to resolve also such sentences in a consistent yet easy-to-annotate manner.

Additionally, in the GPT prompting experiment we experimented with a few different prompts, but did not do exhaustive prompt engineering, and it is possible that with more aggressive prompt engineering GPT can perform better on the task than our results indicate. Similarly for the fine-tuning experiments with T5-large, in which we did some hyperparameter tuning, but not aggressively so.

Ethics Statement

Worker Qualification and Compensation for Annotation. To collect annotations on our dataset, we used Amazon Mechanical Turk (AMT). All

workers had the following qualifications: (1) over 5,000 completed HITs; (2) 99% approval rate or higher; (3) Native English speakers from England, New Zealand, Canada, Australia, or United States. Workers were paid $0.75 per HIT, and on average completed a batch within four hours of work. In addition, $10 was given upon completing a batch (73 HITs), raising the hourly pay to $16.2.

Data Collection and Usage Policy for Annotation. Workers were informed that their annotations would be collected for research purposes and would be used to train and evaluate language-related models, and that the annotations would eventually be made publicly available. Additionally, our task and the annotations collected were of objective nature and did not contain any personal information. Furthermore, all data sources used in the study were publicly available.

9 Acknowledgements

This project received funding from the Europoean Research Council (ERC) under the Europoean Union's Horizon 2020 research and innovation programme, grant agreement No. 802774 (iEXTRACT) and grant agreement No. 677352 (NLPRO). The third author is also funded by a grant from the Israeli Ministry of Science and Technology (MOST), grant number 3-17992.

References

Pranav Anand and Daniel Hardt. 2016. Antecedent selection for sluicing: Structure and content. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1243, Austin, Texas. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

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, Brussels, Belgium. Association for Computational Linguistics.

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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jessica Ficler and Yoav Goldberg. 2016. Improved parsing for argument-clusters coordination. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–76, Berlin, Germany. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

R. A. Hudson. 1973. Conjunction-reduction. *Journal of Linguistics*, 9(2):303–305.

Richard A Hudson. 1989. Gapping and grammatical relations. *Journal of Linguistics*, 25:57–94.

Ray S Jackendoff. 1971. Gapping and related rules. *Linguistic inquiry*, 2(1):21–35.

Yoshihide Kato and Shigeki Matsubara. 2020. Parsing gapping constructions based on grammatical and semantic roles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2747–2752, Online. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3. LDC Catalog No.: LDC99T42, ISBN: 1-58563-163-9, ISLRN: 141-282-691-413-2.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A lightweight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Leif Arda Nielsen. 2004. Verb phrase ellipsis detection using automatically parsed text. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1093–1099, Geneva, Switzerland. COLING.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Dong-woo Park. 2009. On pseudogapping in HPSG. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, pages 425–434, Hong Kong. City University of Hong Kong.

Myung-Kwan Park and Jung-Min Kang. 2007. Multiple sluicing in English. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation*, pages 394–404, Seoul National University, Seoul, Korea. The Korean Society for Language and Information (KSLI).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2019. Controlled crowdsourcing for high-quality qa-srl annotation.

John Robert Ross. 2014. *Gapping and the order of constituents*. De Gruyter Mouton.

Sebastian Schuster, Matthew Lamm, and Christopher D. Manning. 2017. Gapping constructions in Universal Dependencies v2. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 123–132, Gothenburg, Sweden. Association for Computational Linguistics.

Sebastian Schuster, Joakim Nivre, and Christopher D. Manning. 2018. Sentences with gapping: Parsing and reconstructing elided predicates. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1156–1168, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

A.1 Parser

Throughout this project, we use the spacy parser with the out-of-the-box `en_core_web_trf` model.

A.2 Models

T5 Configuration. To ensure reproducibility of our results, we provide the specific configuration of T5-large model used in our study. The model had 770M parameters and was trained with a batch size of 8. The optimizer used was AdamW with a adam_eps value of 1e-8. The maximum input and output length were set to 256 and weight decay was set to 0. The learning rate was set according to the recommendations of the Hugging Face library, with a value of 3e-4. These configurations were used consistently across all variations of the model in our study.

GPT-3 Configuration. In our approach, we use OpenAI's `text-davinci-3` model, a large language model based on the GPT-3 architecture. The temperature is set to 0 and top_p remains 1, resulting in conservative and less-deviant text. The maximum number of tokens generated is set to 256. These parameters have been fine-tuned to control the generated text.

GPT-3 Prompt Format. Here we show a concrete illustration of the instructions provided to GPT-3.

Q: 58.1% of the population described themselves in the 2011 census return as being at least nominally Christian ***<SPLIT> and </SPLIT>*** *0.7% as Muslim with all other religions represented by less than 0.5% each.*
A:
Cannot re-write this sentence.

Q: Federal education assistance offered affordable loans to Americans who wanted to attend college ***<SPLIT> and </SPLIT>*** *money for local schools to ensure that all children received an adequate education.*
A:
Federal education assistance offered affordable loans to Americans who wanted to attend college.
Federal education assistance offered money for local schools to ensure that all children received an adequate education.

Q: He was subsequently asked to repeat the program at the American Asylum for Deaf - mutes in Hartford, Connecticut, ***<SPLIT> and </SPLIT>*** *the Clarke School for the Deaf in Northampton, Massachusetts.*
A:
He was subsequently asked to repeat the program at the American Asylum for Deaf - mutes in Hartford, Connecticut.
He was subsequently asked to repeat the program at the Clarke School for the Deaf in Northampton, Massachusetts.

Q: The plan, a grid with two main axes meeting at a central square ***<SPLIT> and </SPLIT>*** *an additional square in each corner, was based on Thomas Holme's 1682 plan for Philadelphia.*
A:
The plan, a grid with two main axes meeting at a central square, was based on Thomas Holme's 1682 plan for Philadelphia.
The plan, a grid with an additional square in each corner, was based on Thomas Holme's 1682 plan for Philadelphia.

Q: Example alternative schools include Montessori schools, Waldorf schools, Friends schools, Sands School, Summerhill School, The Peepal Grove School, Sudbury Valley School, Krishnamurti schools, ***<SPLIT> and </SPLIT>*** *open classroom schools.*

A.3 Crowdsourcing Task

Here we show the instructions in our crowdsourcing annotation task.

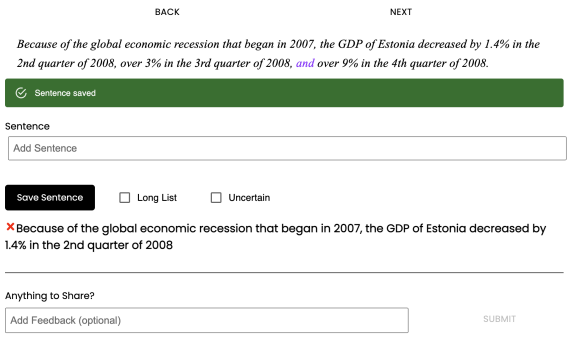


Figure 4: Crowdsourcing task UI

A.4 Crowdsourcing Task Instructions

Rewrite Sentences to Multiple Sentences

You will be shown a sentence such as:

> *As of January 2013, The Times has a circulation of 399,339, The Sunday Times of 885,612, and The New York Times of 9,512,132.*

Please rewrite the sentence to multiple sentences, without the word 'and' and its accompanying commas:

> *1. As of January 2013, The Times has a circulation of 399,339.*
>
> *2. As of January 2013, The Sunday Times has a circulation of 885,612.*
>
> *3. As of January 2013, The New York Times has a circulation of 9,512,132.*

A good rule of thumb is to add the removed 'and' word between the rewritten sentences, and see if the meaning of the original sentence is preserved, as such:

> *As of January 2013, The Times has a circulation of 399,339 and as of January 2013, The Sunday Times has a circulation of 885,612 and as of January 2013, The New York Times has a circulation of 9,512,132.*

Some 'and' words are impossible to remove:

> *Jane has five dollars and fifty cents in her wallet.*
>
> *1. Jane has five dollars in her wallet.*
>
> *2. Jane has fifty cents in her wallet.*

In this case, removing 'and' creates two contradicting sentences, and does not preserve the meaning of the sentence.

In this HIT you will see seven sentences with at least one 'and', please:

1. Rewrite the sentence to multiple sentences with minimal 'and' words and accompanying commas.
2. You are restricted to words (and their past/present/future, singular/plural forms) that appear in the original sentence.
3. If the sentence is a list that calls for **ten rewritings or more,** check the Long List box, and submit the sentence as is.
4. If preserving the meaning of the sentence is impossible, submit the sentence as is.

MORE EXAMPLES

A.5 Dependencies in Verb Nucleus.

As detailed in section 6, a verb nucleus contains a verb and its arguments. While to identify the nucleus root (the verb), we look if their part-of-speech tag is one of ("VB", "VBD", "VBG", "VBN", "VBP", "VBZ"), the rest of the nucleus is defined over the dependency graph:

- subjects – ("nsubj", "nsubjpass", "expl").
- object – ("dobj", "obj", "pobj", "iobj", "attr", "oprd").
- prepositions and their prepositional modifiers – ("prep", "agent") and ("pobj", "pcomp").
- negations – "neg".

Negations are included to account for cases where the model correctly predicts most verb arguments, but fails to account for negation, thus, breaking the original meaning of the sentence. For instance, "The governor urged the public not to panic and to follow his reports closely" is resolved to:

- *The governor urged the public not to panic*
- *The governor urged the public ~~not~~ to follow his reports closely*

A.6 Additional Results

To put the performance of the models in context, we provide results over each conjunct. Moreover, we include exact matching over the sets of sentences, here, punctuation is removed and the sets are assumed to be aligned. See table 4.

Model	$F1_{and}$	$F1_{but}$	$F1_{or}$	F1	Exact Match
$Calibration_1$	4.7	0.0	15.6	5.1	2.4
$Calibration_k$	45.2	30.9	64.9	45.5	2.4
$T5_{10\%}$	59.7	28.9	45.2	55.1	37.3
$T5_{20\%}$	81.2	57.7	72.2	77.7	66.6
$T5_{30\%}$	82.9	60.4	74.2	79.5	69.1
$T5_{40\%}$	**84.3**	70.9	76.3	82.1	72.4
$T5_{50\%}$	83.5	75.7	74.2	81.8	72.6
$T5_{60\%}$	82.6	76.2	76.9	81.3	72.7
$T5_{70\%}$	82.8	76.4	76.7	81.6	73.8
$T5_{80\%}$	83.7	77.7	76.3	**82.3**	73.9
$T5_{90\%}$	82.9	77.5	75.3	81.6	72.6
$T5_{100\%}$	82.6	**79.3**	**82.4**	82.2	**74.9**
$GPT3$	73.5	65.4	70.4	72.3	53.1

Table 4: F1 performance of all models based on their conjunction. For context, an exact match over the sets of sentences is provided.

A.7 Additional Examples of Patterns Indicating Suspicious Sentences

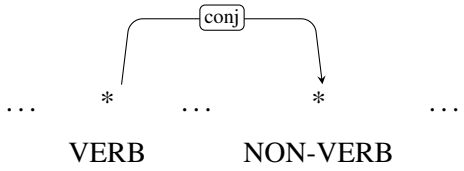
In collecting sentences, we broadly look for three types of "suspicious" structures:

- ***part-of-speech mismatch*** – two words with a different part-of-speech are linked by a *conj*.
- ***dependency relation mismatch*** – an inconsistency between a word's part-of-speech tag and its relations to its dependents.
- ***subtree mismatch*** – two words with the same part-of-speech, but different subtrees.

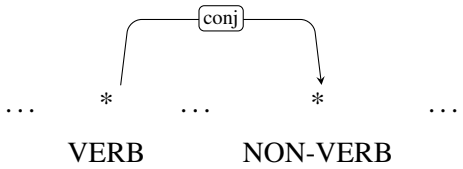
While most patterns involve more than one suspicious structure, we group the examples to match the above list.

The full sentences demonstrating these patterns can be seen below. We denote common-nouns, proper-nouns, adjectives, and numerical values with "NON-VERB", and a verb or an auxiliary verb, with "VERB". When a pattern checks for one of few possible relations between two words, we use "/" to separate them (e.g., *advcl/xcomp* indicates the pattern accepts an *advcl* or an *xcomp* relation). The relation *any* indicates the pattern accepts any type of relation between two words, and *obj* indicates the pattern accepts any type of object.

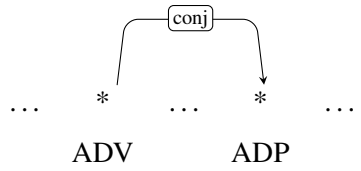
A.7.1 Part-of-speech Mismatch



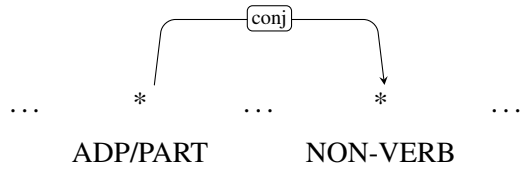
Example sentence: Koreans made up 1.2% of the city's population, and Japanese 0.3%.



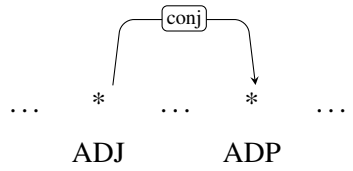
Example sentence: Ten chapters are devoted to body issues and how to cover them.



Example sentence: Desormeaux has won the Preakness twice: once aboard Real Quiet in 1998 and again 10 years later on Big Brown.

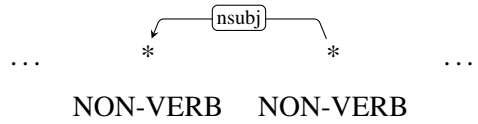


Example sentence: Tell us in the comments below or @CNNFilms on Twitter.



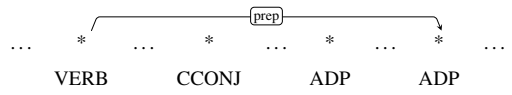
Example sentence: Southwest said all customers were safe and at the terminal.

A.7.2 Dependency Relation Mismatch

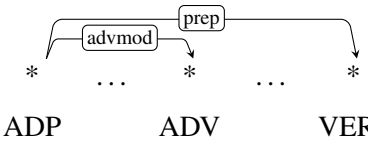


Example sentence: To idealists, spirit or mind or the objects of mind are primary, and matter secondary.

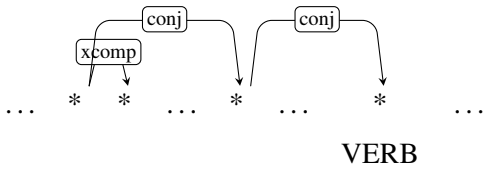
A.7.3 Subtree Mismatch



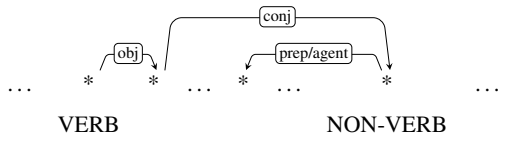
Example sentence: John was born to Henry II of England and Eleanor of Aquitaine on 24 December 1166.



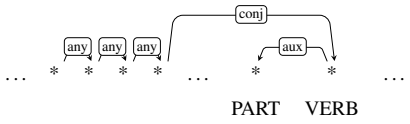
Example sentence: From the 1880s onward neighbourhoods such as Oud- wijk, Wittevrouwen, Vogelenbuurt to the East, and Lombok to the West were developed.



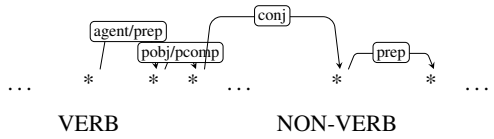
Example sentence: 19 soldiers, policemen reported wounded, and some attackers killed, wounded or captured.



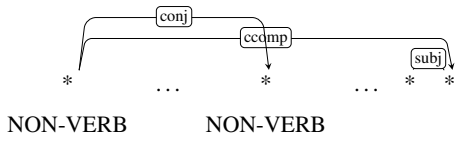
Example sentence: You send out these sound waves, and when they bounce off of objects, the reflection of the waves tells you – or in this case, the animal – where the objects are.



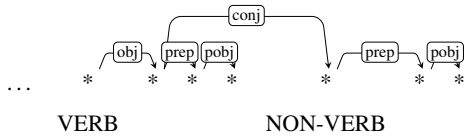
Example sentence: Some runners started raising money for charity or to help with relief efforts.



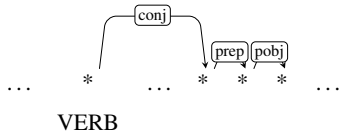
Example sentence: Every day, someone new is introduced to the hardships of wartime military service or the horrors of combat.



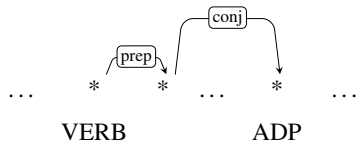
Example sentence: Progress in the Business District but lingering blight in poorer neighborhoods, he says.



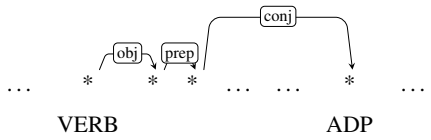
Example sentence: In 1995, material costs were 30 cents for the jewel case and 10 to 15 cents for the CD.



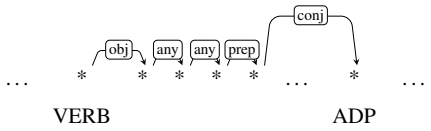
Example sentence: Neesham would make 85 from 80 and Kane Williamson a more considered 54 from 98 as Sri Lanka toiled.



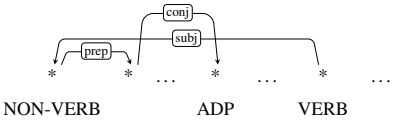
Example sentence: It is also used in woodcut printmaking, and for engraving.



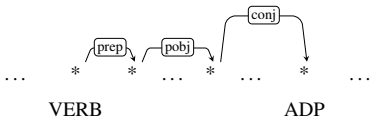
Example sentence: This is The Joker's war on Batman and even more so, on his family.



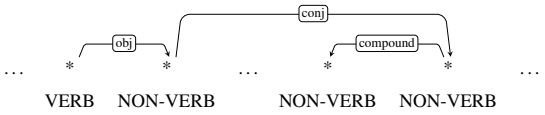
Example sentence: They've been major players in the uprisings in Yemen and in Syria.



Example sentence: Government control of the economy and of expression is much reduced, he says.



Example sentence: They concentrated in trade, services, and especially in money lending.



Example sentence: The meteor show is entertainment for most, but a research chance for NASA.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☒ A2. Did you discuss any potential risks of your work?
Our work collects data from existing well-known public datasets and presents a task to resolve verbal omissions in coordination sentences, we are not aware of a potential risk resulting from it

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and 1 Introduction

☑ A4. Have you used AI writing assistants when working on this paper?
We used it throughout the paper to rephrase the content we already wrote. Specifically, we used ChatGPT with the prompt: "Rephrase in a concise and clear manner"

B ☑ Did you use or create scientific artifacts?

5 Conjunct Resolution Dataset

☑ B1. Did you cite the creators of artifacts you used?
4 Data Collection Process

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All the resources we used are under a license that allows their use for the purpose we used them for.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
1 Introduction and Ethics Statement

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Ethics Statement

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Ethics Statement, 5 Conjunct Resolution Dataset

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
5 Conjunct Resolution Dataset

C ☑ Did you run computational experiments?

Experiments

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
7 Experiments and Appendix We did not discuss computational budget and computing infrastructure, as we did not conduct extensive training

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
7 Experiments

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
appendix

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

4 Data Collection Process

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
4.2 Annotation and Curation

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 4.2 discusses how we recruited, the "ethics statement" discusses pay

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
ethics statement

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
ethics statement