LARP 2025

Proceedings of the 2025 CLASP Conference on Language models and RePresentations

Editors: Nikolai Ilinykh, Erik Lagerstedt, Mattias Appelgren

Gothenburg, Sweden and online
8–9 September 2025

CLASP centre for linguistic theory and studies in probability

CLASP Papers in Computational Linguistics, Volume 7
University of Gothenburg

CLASP Conference Proceedings, Volume 4

Front-cover art: Erik Lagerstedt with the help of GPT-4.5.

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISSN 2002-9764
ISBN 979-8-89176-249-7

Preface

We are delighted to welcome you to the CLASP conference on Language Models and RePresentations or **LARP 2025**! This volume consists of the archival papers presented at LARP, held at the Department of Philosophy, Linguistics and Theory of Science (FLoV), University of Gothenburg on September 8 – 9, 2025. The purpose of this conference was to bring together researchers in computational linguistics, artificial intelligence, and their intersections to discuss ideas on how language can be represented, and how computational language systems can both integrate neural (sub-symbolic) and symbolic representations. The conference covers areas such as computational linguistics, machine learning, artificial intelligence, natural language processing, and more.

The recent advances in language technology have been driven by the large language models (LLMs) built using transformers, large architectures, with representations built out of high dimensional feature spaces. These systems have been highly successful, however, much of human reasoning occurs on a symbolic level following the rules of logic, mathematics, or other systems. Classical AI was focused on symbolic systems, creating expert systems, planners, and search algorithms which manipulated systems, not tensors. There is a growing interest in the idea that these two methods can be combined in order to take advantage of the strengths of each. Many questions arise around these topics. How can neuro-symbolic architectures be created, what are the benefits and problems with them? Can these systems be used to create more explainable machine learning models? Can logical constraints imposed on neural networks increase both explainability, safety, and control over those models? Can automated reasoning provide human interpretable rationals for decisions? LARP invited papers on these topics and more. Accepted papers and invited talks included topics ranging from evaluating the reasoning capabilities of LLMs, bridging the gap between symbolic and neural approaches, abstractions for AI problem solving, to specific implementations of neuro-symbolic and reasoning systems. The conference, and by extension these proceedings, is a discussion about these related topics which examine various approaches and how they can mutually inform each other.

The event included 7 oral talks with presentations of 5 accepted peer-reviewed papers, including 1 archival short paper and 4 archival long papers. The event also had 3 invited keynote talks and a panel discussion. We would like to thank all our contributors, programme committee members, reviewers and volunteers, with special thanks to CLASP for organising the hybrid conference and the Swedish Research Council for funding CLASP.

Nikolai Ilinykh, Erik Lagerstedt, and Mattias Appelgren

Gothenburg, Sweden

September 2025

Organizing Committee

Program Chairs

Nikolai Ilinykh, University of Gothenburg, Sweden
Mattias Appelgren, University of Gothenburg, Sweden
Erik Lagerstedt, University of Gothenburg, Sweden

Local Arrangements Chairs

Susanna Myyry, University of Gothenburg, Sweden
Mattias Appelgren, University of Gothenburg, Sweden

Program Committee

Robin Cooper	University of Gothenburg
Alessandra Zarcone	Technische Hochschule Augsburg
Asad B. Sayeed	University of Gothenburg
Richard Johansson	Chalmers University of Technology and University of Gothenburg
Sharid Loáiciga	University of Gothenburg, Sweden
Maxime Amblard	Université de Lorraine
Bill Noble	Göteborg University and University of Gothenburg
Andy Lücking	Johann Wolfgang Goethe Universität Frankfurt am Main and Université Paris Diderot
Sandro Pezzelle	University of Amsterdam
Katrin Erk	University of Texas, Austin
Elisabetta Jezek	University of Pavia
Shalom Lappin	University of Gothenburg and Queen Mary University of London and King's College London
Nikhil Krishnaswamy	Colorado State University
Joakim Nivre	Uppsala University

Keynote Talk

PACE: Procedural Abstractions for Communicating Efficiently

Moa Johansson
Chalmers University of Technology
2025-09-08 09:30:00 – Room: **J222**

Abstract: A central but unresolved aspect of problem-solving in AI is the capability to introduce and use abstractions, something humans excel at. Work in cognitive science has demonstrated that humans tend towards higher levels of abstraction when engaged in collaborative task-oriented communication, enabling gradually shorter and more information-efficient utterances. In this talk, I will describe a neuro-symbolic method for introducing such abstractions called PACE. On the symbolic side, we draw on work from library learning in program synthesis for proposing abstractions. We combine this with neural methods for communication and reinforcement learning, via a novel use of bandit algorithms for controlling the exploration and exploitation trade-off in introducing new abstractions. Accepted for CogSci 2025 (oral), preprint: https://arxiv.org/abs/2409.20120

Bio: Moa Johansson is an Associate Professor in the Data Science and AI division at Chalmers University of Technology. She is interested in neuro-symbolic AI: the combination of neural machine learning methods and symbolic methods from e.g. theorem proving and program synthesis. Her group works on applications in maths and reasoning, cognitive science, and language.

Keynote Talk

On Retrieving & Reasoning LLMs: Myths, Merits, and How to Move Forward

Dan Roth
University of Pennsylvania and Oracle
2025-09-08 15:00:00 – Room: **J222**

Abstract: The rapid progress made over the last few years in generating linguistically coherent natural language has blurred, in the mind of many, the difference between natural language generation, understanding, knowledge retrieval and use, and the ability to reason with respect to the world. Nevertheless, reliably and consistently supporting high-level decisions that depend on natural language understanding and heterogenous information retrieval is still difficult, mostly, but not only, since most of these tasks are computationally more complex than language models can support. I will discuss some of the challenges underlying reasoning and information access and argue that we should exploit what LLMs do well while delegating responsibility to special purpose models and solvers for decision making. I will present some of our work in this space, focusing on supporting reasoning and information access via neuro-symbolic methods.

Bio: Dan Roth is the Eduardo D. Glandt Distinguished Professor at the Department of Computer and Information Science, University of Pennsylvania and the Chief AI Scientist at Oracle. Until June 2024 Dan was a VP/Distinguished Scientist at AWS AI. In his role at AWS Roth led over the last three years the scientific effort behind the first-generation Generative AI products from AWS, including Titan Models, Amazon Q efforts, and Bedrock, from inception until they became generally available. Dan is a Fellow of the AAAS, ACM, AAAI, and ACL. In 2017, Dan was awarded the John McCarthy Award; he was recognized for "for major conceptual and theoretical advances in the modeling of natural language understanding, machine learning, and reasoning". He has published broadly in natural language processing, machine learning, knowledge representation and reasoning, and learning theory, was the Editor-in-Chief of the Journal of Artificial Intelligence Research (JAIR) and has served as a Program Chair and Conference Chair for the major conferences in his research areas. Roth has been involved in several startups; most recently he was a co-founder and chief scientist of NexLP, a startup that leverages the latest advances in Natural Language Processing, Cognitive Analytics, and Machine Learning in the legal and compliance domains. NexLP was acquired by Reveal. Dan received his B.A Summa cum laude in Mathematics from the Technion, Israel and his Ph.D. in Computer Science from Harvard University in 1995.

Keynote Talk

Reasoning with Large & Small Models: Bridging Symbolic and Neural Approaches

Vaishak Belle
University of Edinburgh
2025-09-09 10:00:00 – Room: **J222**

Abstract: This talk explores the intersection of large language models (LLMs) and reasoning systems, with a focus on addressing fundamental challenges in developing correct and reliable systems. We'll examine our work on augmenting LLMs with external "symbolic executors", creating hybrid architectures that leverage the strengths of both paradigms. The presentation will then talk about how LLMs represent and manipulate beliefs - standing for interactions with human or artificial users. We'll also discuss a few considerations for agentic pipelines, and how these sit with the broader paradigm of agent modelling, which has a long history in AI. We'll preface this development by first briefly reviewing the paradigm of neuro-symbolic AI, and emergent ideas such as loss functions and neural program induction.

Bio: Dr Vaishak Belle (he/him) is a Chancellor's Fellow and Reader at the School of Informatics, University of Edinburgh. He is an Alan Turing Institute Faculty Fellow, a Royal Society University Research Fellow, and a member of the RSE (Royal Society of Edinburgh) Young Academy of Scotland. He was previously at KU Leuven (Belgium), University of Toronto (Canada), Aachen University of Technology (Germany) and University of Trento (Italy). At the University of Edinburgh, he directs a research lab on artificial intelligence, specialising in the unification of logic and machine learning, with a recent emphasis on explainability and ethics. He has given research seminars at academic institutions such as MIT and Oxford, tutorials at AI conferences, and talks at venues such as Ars Electronica and the Samsung AI Forum. He has co-authored close to 120 peer-reviewed articles on AI, at venues such as IJCAI, UAI, AAAI, MLJ, AIJ, JAIR, AAMAS, and along with his co-authors, he has won the Microsoft best paper award at UAI, the Machine learning journal best student paper award at ECML-PKDD, and the Machine learning journal best student paper award at ILP. In 2014, he received a silver medal by the Kurt Goedel Society. He has served on the senior program committee/area chair of major AI conferences, co-chaired the ML track at KR, among others, and as PI and CoI secured a grant income of close to 8 million pounds. Recently, he has consulted with major banks on explainable AI and its impact in financial institutions.

Table of Contents

Simple Morphology, Complex Models: A Benchmark Study and Error Analysis of POS Tagging for Martinican Creole
Ludovic Mompelat 1

EventHopNLI: A Functional Dataset for Systematically Diagnosing Logical Failures in LLM Temporal Reasoning
Ved Mathai and Janet B. Pierrehumbert 11

Combining Information State Update, Harel Statecharts and LLMs for controllable and flexible Conversational AI
Vladislav Maraev, Alexander Berman and Staffan Larsson 28

Towards Neuro-Symbolic Approaches for Referring Expression Generation
Manar Ali, Marika Sarzotti, Simeon Junker, Hendrik Buschmeier and Sina Zarrieß 38

Extracting a Prototypical Argumentative Pattern in Financial Q&As
Giulia D'Agostino, Michiel Van Der Meer and Chris Reed 51

Simple Morphology, Complex Models: A Benchmark Study and Error Analysis of POS Tagging for Martinican Creole

Ludovic Mompelat
Department of Modern Languages and Literatures
University of Miami
Miami, FL, USA
lvm861@miami.edu

Abstract

Part-of-speech (POS) tagging is a foundational task in NLP pipelines, but its development for Creole languages remains limited due to sparse annotated data and structural divergence from high-resource languages. This paper presents the first POS tagging benchmarks for Martinican Creole (MC) as well as a linguistically motivated evaluation framework, comparing three fine-tuned transformer-based models (mBERT, XLM-Roberta, and CreoleVal). Rather than focusing solely on aggregate metrics, we perform detailed error analysis, examining model-specific confusion patterns, lexical disambiguation, and out-of-vocabulary behavior. Our results yield F1 scores of 0.92 for mBERT (best on the X tag and connector distinctions), 0.91 for XLM-Roberta (strongest on numeric tags and conjunction structures), and 0.94 for CreoleVal (leading on both functional and content categories and lowest OOV error rate). We propose future directions involving model fusion, targeted and linguistically motivated annotation, and reward-guided Large Language Models data augmentation to improve our current tagger. Our linguistically grounded error analysis for MC exposes key tagging challenges and demonstrates how targeted annotation and ensemble methods can meaningfully boost accuracy in under-resourced settings.

1 Introduction

Despite significant progress in multilingual language modeling (Qin et al., 2024; Huang et al., 2024), natural language processing (NLP) for Creole languages remains underdeveloped. This is largely due to the scarcity of annotated resources and the unique linguistic features of Creoles such as morphosyntactic restructuring and realignment (Mufwene, 2013), and frequent code-switching/code-mixing (Vaillant, 2023) which challenge existing models trained on high-resource languages (Mompelat et al., 2022). This paper introduces the first benchmark part-of-speech (POS) tagging dataset for Martinican Creole (MC) and presents a comparative evaluation of three transformer-based models: XLM-Roberta (Conneau et al., 2019), mBERT (Devlin et al., 2018), and CreoleVal (Lent et al., 2024), the latter being a recent adaptation specifically designed for Creole NLP tasks.

Accurate POS tagging is an important aspect of NLP pipelines, directly affecting the performance of downstream applications such as dependency parsing (Zhou et al., 2020), machine translation (Hlaing et al., 2022), and information extraction (Chiche and Yitagesu, 2022). Yet for languages like MC, both the lack of training data and the linguistic divergence from typologically dominant languages like French present ongoing obstacles. In previous work, Mompelat et al. (2022) demonstrated that cross-lingual transfer from French improved syntactic parsing performance for MC, but it also introduced cascading errors in the POS tagging stage, therefore highlighting the need to treat POS tagging as a distinct problem.

Rather than focusing solely on accuracy metrics, this paper takes a linguistically informed approach to model evaluation. We analyze tagging errors across the three models, with particular attention to phenomena such as lexical and syntactic ambiguity, as well as out-of-vocabulary (OOV) words. By examining classification reports, confusion matrices, support-F1 dynamics, and error patterns linked to specific linguistic features, we reveal model-specific strengths and weaknesses that would be obscured by aggregate scores alone.

This analysis not only benchmarks current POS tagging performance for MC but also informs future work on multi-model strategies, such as weighting predictions across models or implementing multi-task learning. It also guides annotation efforts, helping to determine which linguistic phenomena and patterns most merit attention in resource-constrained settings. Ultimately,

Proceedings of the 2025 CLASP Conference on Language models And RePresentations (LARP), pages 1–10

this study sets a precedent for linguistically motivated model evaluation and resource development for underrepresented languages. As part of a larger project aimed at improving parsing tools for MC and other Creole languages, it illustrates how targeted, linguistically grounded interventions, beginning with POS tagging, can incrementally strengthen performance and serve as a foundation for future applications, including rule-based data augmentation, transfer learning, and hybrid symbolic-neural modeling.

2 Review of the Literature

2.1 Martinican Creole and Its Relation to French

Martinican Creole (MC) is a French-lexified Atlantic Creole spoken predominantly in Martinique. Despite heavy lexical overlap with Modern French, MC exhibits several typological divergences: it is largely isolating, with minimal inflectional morphology for tense, aspect, or agreement (see tense marker *ké* and mood marker *ka* in example (1)), and it uses post-posed determiners and possessives (see definite marker *an* in example (1)).

French and Creole have historically been described to coexist in a diglossic relationship, with French historically associated with institutional and high-prestige functions, and Creole with informal and oral domains. However, this diglossic split is starting to break down and we see both languages sharing space in virtually all functions of society (Prudent, 1981; Bernabé, 1983; Managan, 2016). This constant language contact situation in all functions results in the two languages to often be mixed within the same discourse. This functional break down between MC and French leads to prevalent code-mixing and code-switching, necessitating models that are robust to lexical variation. This close genealogical and historical link justifies cross-lingual transfer approaches, yet the subtle orthographic and phonological differences between French and MC, together with MC's unique syntactic patterns, pose challenges for direct transfer of standard POS tagsets.

2.2 Polyfunctionality in MC versus Homonymy

In MC, many closed-class items exhibit *polyfunctionality*: the capacity to serve multiple grammatical categories without any overt morphological change. We follow the terminology of Wang et al. (2021) whereby words that have more than one part of speech are called *polyfunctional words*, while words with only one part of speech are called *monofunctional words*. Polyfunctionality differs from polysemy, since all senses of a polysemous word may belong to the same POS category.

In our MC corpus, several high-frequency items display PF > 1, driving much of the POS-tagging ambiguity. Below are illustrative examples for the marker *ki* (PF = 3):

(1) KI = PRON, SCONJ, DET

a. Sé pa mwen **ki** pou réparé
COP not me who must fix
lektrisité [...]
electricity
'I'm not the one **who** must fix the electricity [...]' PRON

b. chak moun-an ja ka di **ki**
each person-DET already PROG say that
yo pa dakò.
3PL not agree
'Each of them are already saying **that** they disagree' SCONJ

c. Jik **ki** tan nou ké asepté yo
until which time we FUT accept 3PL
fè nou wont kon sa ?
make 2PL shame like that
'Until when will we let them embarass us like that?' DET

Similarly, *kon* alternates among coordinating conjunction (CCONJ), subordinating conjunction (SCONJ), and adposition (ADP) as shown in example (2).

(2) KON = CCONJ, SCONJ, ADP

a. Rad-maré anni balié lakot Atlantik
tidal-wave only sweep coast atlantic
kon lakot karayib
as_well_as coast caribbean
'The tidal wave only swept the Atlantic Coast **as well as** the Caribbean Coast' CCONJ

b. Kon di Kolo, 'si ou pa ri yo,
like say Kolo, 'if 2SG not laugh 3PL,
yo ké ri'w'.
3PL FUT laugh'2PL
'**Like** Kolo says, "If you don't laugh at them, they'll laugh at you' SCONJ

c. Jik ki tan nou ké asepté yo
until which time we FUT accept 3PL
fè nou wont **kon** sa ?
make 2PL shame like that
'Until when will we let them embarass us like that?' ADP

By contrast, true *homonymy* involves two or more unrelated lexical entries sharing form but with distinct, non-overlapping etymologies and meanings (e.g. MC *sé*$_1$ "it is" from French "c'est" acting as a copula vs. *sé*$_2$ from French "ces" acting as a plural marker).

(3) Kataloy, **sé** réjion ki pi rich adan tout
Catalonia, COP region that most rich in all
sé réjion l'Espay la
PLUR region Spain DET
'Catalonia, **it's** the richest region in all **the** region**s** of Spain' COP/DET

In example (3), we see that the homonym *sé* serves as a determiner and a copular predicate marker.

The frequent polyfunctionality of closed-class words is typical of isolating Creoles and underscores the importance of a UD-compliant annotation that preserves each usage and motivates our linguistically informed error analysis.

2.3 Neural Approaches to POS Tagging and Its Role in NLP

Part-of-speech (POS) tagging is an valubale task in natural language processing, as it provides essential grammatical structure that enables models to make more informed linguistic predictions. For example, Hlaing et al. (2022) showed that POS tags can be leveraged as syntactic signals to improve neural machine translation in low-resource language pairs.

While POS tagging systems for high-resource languages now achieve near-human accuracy, their development and evaluation are relatively straightforward due to the abundance of annotated corpora. In contrast, the task assumes greater importance in low-resource settings, where POS tags may be the only structured representation available. They can serve as scaffolding for downstream tasks such as dependency parsing (Mompelat et al., 2022) or Machine Translation, and as a way to stabilize training in scenarios where full syntactic or semantic annotations are lacking.

However, POS tagging for low-resource languages faces challenges on multiple fronts. First, the scarcity of labeled data makes it difficult for supervised models to learn robust tag distributions or to use unsupervised training methods. Multilingual languages models such as mBERT and XLM-R have become widely adopted as they excel in transferring knowledge from high-resource languages (like English) to low-resource ones, even without parallel data (Pires et al., 2019). However, their performance is uneven across languages, particularly when faced with typological distance, orthographic variability, or underrepresentation in pretraining data. Many low-resource languages, especially Creoles, diverge typologically from the languages on which these multilingual models have been trained as they often exhibit minimal inflection, fluid category boundaries, and weak morphological cues that may be underrepresented in current multilingual models (Hedderich et al., 2020).

These challenges warrant the need for typological and linguistically-aware modeling choices, whether neural, symbolic, or hybrid, using insights from more detailed parsing error analysis.

2.4 Approaching POS Tagging for Creole Languages

Despite advances in low-resource NLP, Creole languages remain severely underrepresented in the development of computational resources and tools. Only a handful of projects have produced annotated corpora for Creoles, and among these, part-of-speech (POS) tagging has received limited focused attention. The recent CreoleVal benchmarks (Lent et al., 2024) introduced a multilingual POS tagging dataset for Haitian, Mauritian, and 26 other Creoles, alongside a transformer-based model trained on these data. We note that MC is not included in the dataset from the CreoleVal project. Therefore, although this represents a significant step forward for Creole languages, the datasets remain modest in size, often domain-restricted, unevenly distributed across tasks, and the model necessitates fine-tuning for MC.

Prior work by Mompelat et al. (2022) proposed a dependency parser for MC that leveraged French as a support language via cross-lingual transfer. Although this approach improved parsing performance, it also revealed notable shortcomings in the POS tagging layer. The overall accuracy scores for dependency parsing relying on TAG embeddings showed evidence that differences in morphosyntactic structure between French and MC may be the source of the tagging inconsistencies. These results underscore the need to treat POS tagging in Creole languages as a task in its own right, rather than a secondary artifact of parsing models trained on other languages.

However, relying solely on coarse metrics such as overall accuracy or macro-F1 scores, especially in structurally complex or data-scarce settings has its limitations. In this paper, we propose a more

comprehensive evaluation framework aimed at providing a deeper understanding of model behavior, uncovering patterns of misclassification that would otherwise be obscured by aggregate performance metrics. As Schöffel et al. (2025) show in their study of Old Occitan, low-frequency categories such as interjections, proper nouns, or borrowed terms are especially prone to misclassification by neural taggers, even when macro metrics remain high. In doing so, it also helps identify gaps in the annotated data and motivates more targeted annotation strategies.

2.5 Error Analysis in POS Tagging

Error analysis is a critical tool for understanding the systematic failures of NLP models, especially in contexts where structural ambiguity and low-resource constraints compound the difficulty of robust language modeling. Studies such as Garcia and Gamallo (2010) have demonstrated that error-driven rule-based correction can significantly improve tagging accuracy, particularly when errors are concentrated in predictable linguistic contexts such as confusion between adjectives and nouns, or misinterpretation of closed-class items like determiners, adpositions, and subordinating conjunctions. This suggests that error analysis not only diagnoses model weaknesses but can actively guide remediation strategies through symbolic or hybrid interventions.

In our study, we build on these precedents by introducing a comprehensive statistical error analysis of POS tagging in MC. This includes confusion matrix interpretation, per-tag F1 tracking, identification of homonymous/polysemous tokens, Out-Of-Vocabulary (OOV) error reports, and a support-F1 LOESS analysis to capture the interaction between tag frequency and performance. Our proposed metrics are motivated by MC's high levels of lexical ambiguity/polyfunctionality, frequent borrowing, and minimal morphological marking. To date, no comprehensive error analysis has been conducted for POS tagging in MC or other French-lexified Creoles.

Importantly, our goal is not only to better understand model behavior, but to use these findings to inform the design of targeted data augmentation strategies via large language models. In contrast to Schöffel et al. (2025) who use LLMs directly for tagging evaluation, we intend to leverage them as generative and augmentation tools, guided by linguistic insights extracted through error analysis.

3 Methodology

Our approach is organized into two main phases: first, the creation of a POS tagging benchmark for Martinican Creole (MC) using transformer-based models, and second, the creation of an evaluation framework for fine-grained error analysis to inform linguistically and symbolically future annotation efforts, automatic data augmentation methods and new, hybrid strategies for the development of NLP tools for MC and other low resource languages.

3.1 Dataset and Annotation Process

The dataset used in this study combines the depedency parsing corpus introduced in Mompelat et al. (2022), containing 236 manually dependency-annotated sentences in MC and from which we only extracted the POS annotations, with 298 additional sentences annotated for POS for this project. Although ideally multiple native-speaker linguists would adjudicate, all 298 new sentences were annotated by the author, a heritage speaker of MC with formal training in Universal Dependencies schemas, due to a severe lack of Martinican UD experts, typical for underrepresented languages. To ensure quality, we conducted two full consistency passes over the data and spot-checked ambiguous tokens against a small panel of native speakers. The full dataset thus comprises 534 sentences, 9470 tokens, and 1780 types, making it the most extensive POS-annotated corpus for MC. The data are drawn from online news sources, blogs, and social media, reflecting contemporary usage and the frequent presence of code-mixed French elements. These mixed tokens are essential for capturing the diglossic and bilingual nature of Martinican linguistic practice.

Annotation followed the Universal Dependencies (UD) guidelines for POS tagging (De Marneffe et al., 2021). A key annotation tag concerns the treatment of foreign words, particularly French lexical items. When a French-origin word is syntactically integrated into the MC sentence, functioning as a noun, verb, or modifier, it was tagged according to its grammatical role using standard UPOS categories. However, when the foreign word appeared as a translation equivalent or gloss, not syntactically integrated into the clause structure, it was assigned the X tag, in line with UD conventions for unclassifiable or extragrammatical tokens. Although the "X" tag is often excluded or minimized in evaluation tasks, we chose to preserve it

as a focus in our error analysis since this decision enables us to track annotation ambiguity and model behavior around edge cases, rather than suppressing them from the training and evaluation process.

After initial submission of this paper, we discovered annotation inconsistencies in the MC POS corpus. We have corrected these and fixed a random seed (seed=42) for our train/dev/test splits. All results below reflect this finalized dataset.

3.2 Model Selection and Experimental Design

To evaluate POS tagging performance on MC, we selected three transformer-based models: 1) XLM-Roberta (xlm-roberta-base), a multilingual transformer pretrained on CommonCrawl data containing 100 languages, 2) mBERT (bert-base-multilingual-cased), a widely used multilingual model pretrained on the BooksCorpus and Wikipedia, and 3) CreoleVal, a domain-adapted XLM-R transformer model fine-tuned on the CreoleVal benchmark (Haitian, Mauritian, and 26 other Creoles; Lent et al. 2024). Its subword vocabulary and pre-training did *not* include any MC data, making it directly comparable to XLM-R and mBERT for MC POS tagging. Although CreoleVal and XLM-R share the same tokenizer and subword vocabulary, their parameter distributions diverge during CreoleVal's in-domain fine-tuning. By adapting XLM-R weights on a multilingual Creole benchmark (Haitian, Mauritian, etc.), CreoleVal becomes highly specialized on core Creole patterns, boosting closed-class and function-word tagging, but can "forget" some of XLM-R's broader multilingual robustness. In practice, this leads XLM-R to outperform CreoleVal on proper nouns and rare connectors: its untouched pre-training retains more general representations for named entities and low-frequency items, whereas CreoleVal's weights have shifted toward the distributions encountered in its fine-tuning data.

Each model was fine-tuned on our annotated dataset using an 80/10/10 split for training (427 sentences), development (53 sentences), and testing (54 sentences). The splits also broadly preserve the overall distribution of common POS tags.

3.3 Evaluation and Error Analysis

To assess the behavior and weaknesses of our POS tagging models in a linguistically informed manner, we adopt a suite of diagnostic metrics tailored to the structural characteristics of MC.

We begin with the standard classification report shown in Table 1, which provides precision, recall, and F1 scores per tag. We further examine the relationship between data availability and tagging performance through a support-F1 analysis shown in Figure 1. By plotting F1 scores against tag frequency using LOESS smoothing, we can estimate the number of examples required for each tag to achieve reliable performance/information balance that directly informs future annotation priorities. Then, to uncover more nuanced patterns of misclassification, we analyze confusion matrices, shown in Figure 2, that reveal frequent tag-level confusions such as NOUN versus PROPN or ADP versus SCONJ. These confusion patterns are particularly relevant in MC, where the absence of morphological cues often makes syntactic functions harder to disambiguate and where homonymy/polysemy is frequent. To evaluate the model's ability to resolve morphosyntactic ambiguity, we conduct a homonymy/polysemy error analysis by tracking tokens that occur with multiple POS tags in the corpus. Finally, we perform an OOV error analysis, examining how models handle test-set tokens not seen during training. This includes measuring the overall OOV error rate and identifying common misclassification patterns, such as overpredicting NOUN or confusing named entities and borrowed forms.

4 Results & Discussion

4.1 Overall Model Comparison: Tagging Accuracy by Category

We begin our evaluation with overall accuracy: fine-tuned mBERT achieves 92% , XLM-Roberta 91%, and CreoleVal tops at 94%. Table 1 presents per-tag F1 scores on the Martinican Creole (MC) test set.

All three models achieve perfect or near-perfect scores on the most frequent, low-ambiguity classes: PUNCT, PART, and PRON. They also perform strongly on PROPN. By contrast, rare tags such as INTJ (support=2) remain challenging.

Mid– and low–frequency tags reveal clear differentiation. CreoleVal leads on most functional and content categories, while mBERT retains the edge on coordinators and interjections. In sum, CreoleVal delivers the highest overall F1 by excelling on both function- and content-POS tags; mBERT and XLM-Roberta bring complementary strengths on rare or subtle categories, motivating the model-fusion strategies described in Section 4.6.

Table 1: Per-tag F1 scores across models. Highest values per row are in **bold**.

POS Tag	mBERT	XLM-R	CreoleVal
ADJ	0.77	0.77	**0.82**
ADP	0.81	0.74	**0.84**
ADV	0.81	0.82	**0.86**
AUX	0.95	0.95	**1.00**
CCONJ	**0.90**	0.83	0.86
DET	0.90	0.91	**0.95**
INTJ	**0.67**	0.50	0.50
NOUN	0.88	0.89	**0.91**
NUM	0.86	**0.92**	0.89
PART	0.98	0.99	0.99
PRON	0.99	0.97	**1.00**
PROPN	0.93	0.92	0.93
PUNCT	1.00	1.00	1.00
SCONJ	0.74	0.67	**0.77**
VERB	0.94	0.95	**0.96**
X	0.94	0.94	0.94
TOTAL	0.92	0.91	**0.94**

4.2 LOESS F1 vs. Support Analysis

The analysis presented here utilizes Locally Estimated Scatterplot Smoothing (LOESS), a non-parametric regression method, to explore how the number of annotated examples per POS tag (support) relates to model performance (F1-score). LOESS smoothing is particularly beneficial when dealing with limited, unevenly distributed data points.

To derive a practical annotation threshold for each model, we first compute a smoothed F1-vs-support curve via LOESS. Rather than simply picking the support level at which the smoothed curve first crosses our target F1 of 0.90 (which can be susceptible to isolated "spikes" or noise), we identify all contiguous runs of support values where the LOESS-smoothed F1 remains at or above 0.90. We then select the longest such run, namely the largest consecutive region of stable, high performance, then take its minimum support value as our recommended threshold. This ensures the threshold reflects a region where performance truly "stabilizes" above the target, not just a brief local bump.

Although this method only provides an approximate guide (it does not guarantee per-tag minimums in every context), it offers a more robust estimate of how many annotated tokens are needed, on average, before a model can be expected to perform reliably above an F1 of 0.90. These thresholds can then inform future corpus development and annotation planning.

From the LOESS smoothed curves, shown in Figure 1, we observe that models differ significantly in their data requirements for achieving a high-performance threshold. Specifically, the analysis reveals that the CreoleVal model reaches this threshold with fewer annotated instances (support threshold = 54) compared to XLM-R (support threshold = 121) and mBERT (support threshold = 78). Overall, this analysis clarifies the data-driven relationship between annotation volume and model performance, directly informing practical decisions on corpus annotation strategies and model deployment.

4.3 Error Profile per Model : Tag Confusions

In this section, we analyze the confusion matrices and polyfunctionality-driven errors to understand where models tend to confuse tags, and which types of tokens are consistently hard to disambiguate.

The confusion matrices shown in Figure 2 highlight two interrelated sources of error that stem from the structural characteristics of MC (and many other Creoles). First, content words in MC are highly polyfunctional: A single word can freely serve as a NOUN, ADJ or VERB, which leads our models to routinely confuse our ADJ/VERB/NOUN labels. Second, the lack of overt morphological marking on functional tokens makes it difficult to distinguish conjunctions (SCONJ/CCONJ) from prepositional markers (ADP), resulting in systematic SCONJ $\leftrightarrow$ ADP $\leftrightarrow$ CCONJ errors.

Figure 2 shows that when we aggregate off-diagonal confusions among ADJ, NOUN, and VERB, CreoleVal commits only 12 such errors, compared to 16 by mBERT and 16 by XLM-Roberta. The fact that CreoleVal produces the fewest misclassifications here highlights its superior handling of MC's polyfunctional content items as a Creole-specific language model. Additionally, for the triad ADP–CCONJ–SCONJ, mBERT makes the fewest confusions (11 total), with CreoleVal next (12) and XLM-Roberta last (16). This suggests that mBERT's broader multilingual pretraining better preserves subtle distinctions among functional markers, whereas CreoleVal's in-domain adaptation trades away a small amount of this fine-grained discrimination.

Together, these patterns reveal complemen-

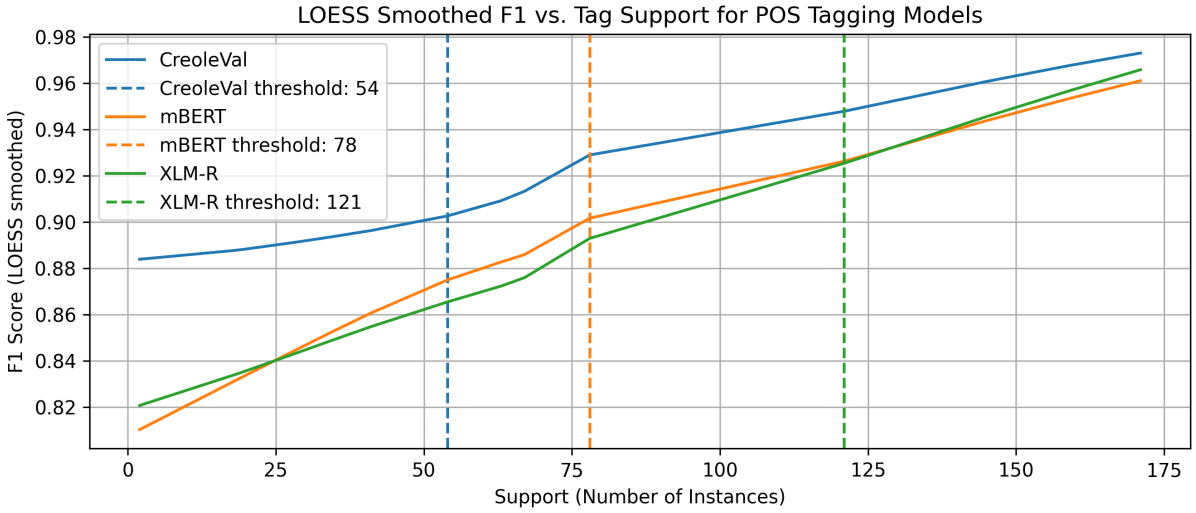


Figure 1: F1-score/support LOESS-smoothed curves for CreoleVal, mBERT, and XLM-Roberta models.

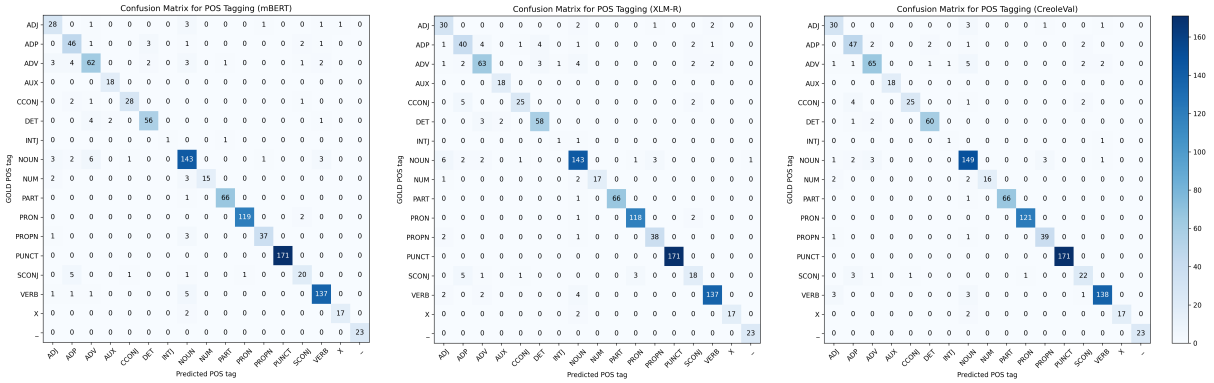


Figure 2: Confusion Matrices for mBERT, XLM-Roberta, and CreoleVal on the Martinican Creole test set.

tary strengths: CreoleVal excels at polyfunctional content-word tagging, mBERT at conjunction/preposition boundaries, and XLM-Roberta lags on both fronts, motivating our fusion strategies to combine their niche advantages in Section 4.6.

4.4 Error Profile per Model : Lexical Ambiguity

Several polyfunctional tokens stand out as persistent error sources, illustrating how MC's category-shifting forms challenge neural taggers. In particular:

- **ki** (30 occurrences): error rates are 3.3% for CreoleVal (1/30), 16.7% for XLM-R (5/30), and 10% for mBERT (3/30). Confusions swap primarily between PRON and SCONJ, with occasional DET misassignments.

- **kon** (6 occurrences): error rates climb to 50% for CreoleVal (3/6), 33.3% for mBERT (2/6), and 66.7% for XLM-R (4/6), reflecting its trifunctional usage as CCONJ, SCONJ, and ADP.

- **pou** (21 occurrences): errors occur at 19% for both CreoleVal and XLM-R (4/21), and 23.8% for mBERT (5/21), with frequent flips between ADP and SCONJ.

Comparing these three confirms a strong and expected correlation between frequency and error rate : the more examples a form has, the more reliably it is tagged. Overall, CreoleVal makes the fewest errors on polyfunctional items, except for *kon*, where its mid-rank performance suggests that exceptionally low support still hampers even the Creole-adapted model.

The homophonous word **sé** ("c'est" vs. "ces", 25 occurrences) exemplifies true homophony rather than just polyfunctionality. Here CreoleVal fully resolves the distinction, while mBERT and XLM-R each mislabels the word on 8% of cases (2/25) as

AUX instead of DET. This difference suggests that in-domain Creole fine-tuning helps models learn language-specific lexical disambiguations that multilingual pretraining alone may miss.

We see that it is these high-frequency polyfunctional and homophonous tokens (and not rare, monofunctional forms) that drive most systematic tagging failures. Remedying them will require richer syntactic context (e.g. dependency relations) or targeted annotation of ambiguous constructions to guide models toward the correct POS distinctions.

4.5 Generalization Limits: OOV Word Behavior

Generalization to unseen vocabulary remains a critical challenge in MC tagging. In our low-resource setting, OOV tokens account for 144 tokens (1.5% of the test set), spanning proper names, phonological variants, idiomatic compounds, and loanwords. CreoleVal mislabels 19 / 144 OOV tokens (13.2%), compared to 23 / 144 (16.0%) for XLM-Roberta and 29 / 144 (20.1%) for mBERT.

A closer look at the misclassified OOV items reveals three dominant error patterns. First, verbs that are morphologically or phonologically similar to the French variants, like **lavé** (*laver*), **payé** (*payer*), **prononsé** (*prononcer*) are persistently mis-tagged as adjectives or nouns. Second, adjectives like **diféran** (*different*) or adverbs like **asé** (*enough*) were frequently flipped between ADJ, ADV, or NOUN. Third, proper-name errors are rare: mBERT alone errs on *Martinique* (PROPN→NOUN), and XLM-R on *guardia* (NOUN→PROPN), underscoring CreoleVal's in-domain fine-tuning advantage for named entities. Finally, foreign words (*galaxie*, *gravitation*) were mis-tagged as NOUN instead of X(2 errors each in CreoleVal and XLM-R, 2 in mBERT), indicating that fully integrated loanwords might confuse the "unclassifiable" category "X" as described in the UD guidelines. Therefore, none of the taggers seem to easily discriminate between fully-integrated borrowed words and foreign insertions. The ability to perform such discrimination is of great importance when dealing with context of enhanced code-switching and code-mixing.

While additional pre-training data can help reduce low-coverage gaps, such large-scale pretraining is often impossible for truly low-resource communities. Our focus here is on fine-tuning existing multilingual models, which offers a more accessible path to improved accuracy. Effective strategies may include explicit linguistically-motivated lexicon augmentation for proper names, enhanced subword tokenization for morphologically complex variants, and hybrid approaches that combine neural tagging with lookup tables for fixed expressions. Only by addressing these structural gaps can we push beyond the generalization limits of our current fine-tuned systems.

4.6 Towards Model Fusion: Leveraging Complementary Strengths

Our analyses confirm that each model brings distinct advantages: CreoleVal delivers the strongest performance on many functional and closed-class tags, leading on DET (0.95 vs. 0.90 vs. 0.91), PART (0.99 vs. 0.98 vs. 0.99), AUX (1.00 vs. 0.95 vs. 0.95), and ADV (0.86 vs. 0.81 vs. 0.82). It also yields the lowest OOV error rate (13.2% vs. 16.0% vs. 20.1%).

XLM-Roberta excels on numeric tags (NUM = 0.92 vs. 0.89 vs. 0.86), reflecting its potential for robust subword representations for rare morphosyntactic constructions. CreoleVal and mBERT both top the named-entity tag (PROPN = 0.93 vs. 0.93 vs. 0.92), while CreoleVal and XLM-R share the highest X performance (0.94 vs. 0.94 vs. 0.92), demonstrating superior handling of fully foreign insertions. mBERT remains the most balanced generalist, with particularly strong scores on CCONJ (0.90), and PRON (0.99).

This clear complementarity suggests several fusion strategies. First, per-token weighted voting, where each model's tag-specific validation F1 determines its vote weight, could improve accuracy on challenging categories like SCONJ (F1 0.67–0.77). Second, per-tag delegation, assigning each POS to its top specialist, would directly leverage CreoleVal's mastery of function words, XLM-Roberta's numeric proficiency, and mBERT's connector expertise. Third, a multi-task architecture combining all three contextual embeddings into a unified classifier may learn to trust each representation dynamically.

Beyond accuracy gains, this fusion approach also informs annotation priorities: SCONJ tags and highly polyfunctional items (e.g. *ki*, *kon*) still incur error rates up to 66.7% under XLM-R. Targeted annotation or data augmentation for these high-ambiguity forms, and enriching their syntactic contexts with dependency relations, may amplify the benefits of any ensemble, ensuring future MC

taggers combine the best of each model's strengths.

4.7 Linguistic Insights for Targeted Annotation and Data Augmentation

This study offers key linguistic takeaways that can directly inform future annotation priorities and data augmentation strategies for MC and other contact-influenced low-resource languages. The errors made by even the strongest taggers are not arbitrary; they reveal systematic patterns shaped by the typological characteristics of Creole morphosyntax.

First, the fluidity of Creole categories where the same form can function as verb, noun, adjective, or connective without overt inflection emerges as the root cause of many systematic confusions. Rather than dispersing annotation effort evenly, we should concentrate on sentences that illustrate this fluidity. In practice, this means mining the corpus (or synthetic data) for contexts where high-ambiguity items like *ki*, *kon*, and *pou* appear in each of their roles, then creating compact annotation batches that cover all readings of a single token. By focusing scarce human effort on these multifunctional "edge cases" we ensure the model sees the precise contextual cues needed to resolve category overlap, rather than redundantly tagging unambiguous examples.

Second, our LOESS-informed support thresholds identify which POS tags remain under-supported even in the best model. In total, 10 of the 16 UPOS categories fall below the 54-instance mark required for stable F1 under CreoleVal: specifically ADJ (35), AUX (18), CCONJ (32), INTJ (2), NUM (20), PROPN (41), SCONJ (28), X (19), alongside marginal cases of ADP (54) and ADV (78). A targeted annotation drive that brings each of these tags up to at least 54 examples would allow significant gains rather than spreading effort at random across already well-learned categories.

Third, the remaining OOV errors on phonological variants, loanwords, and code-switched items spotlight the need for lexicon-aware augmentation. Instead of relying on larger pre-training corpora, which may be infeasible for MC, we can inject synthetic examples of rare compounds (e.g. *alé-vini*), phonetic spellings (*lwen*, *vré*), and integrated borrowings (*vulgaire*) into the training data. Using few-shot LLM prompting, guided by our linguistic error profile, we can generate minimal pairs that contrast these forms in their correct contexts, helping the tagger anchor them to the right POS class. This is to be tested in a near-future experiment.

Finally, these insights advocate for an active-learning loop in which model disagreement and low-confidence predictions drive both annotation selection and augmentation design. By letting the taggers themselves flag the most contentious tokens, we turn our error analyses into a continuous feedback mechanism. Over successive cycles, this linguistically informed, resource-efficient strategy may allow us to deliver MC taggers that not only achieve higher accuracy but also demonstrate a deeper understanding of the language's typological complexity.

5 Conclusion

In this work we have presented the first fine-tuned transformer models for part-of-speech tagging of Martinican Creole (MC) along with a linguistically grounded evaluation framework. Building on a UD-style POS corpus of 534 sentences, we compared mBERT, XLM-Roberta, and CreoleVal. Our results yield accuracy scores of 0.92 for mBERT, 0.91 for XLM-Roberta, and 0.94 for CreoleVal. Beyond scores, our analyses uncover the linguistic dimensions of tagging difficulty in MC: polyfunctional word classes, blurred syntactic boundaries, and code-switching all contribute to high tag ambiguity.

The complementary model strengths we observe suggest lightweight ensemble strategies: for example, per-tag delegation can assign each POS to its specialist model; weighted voting can boost performance on the most ambiguous categories; and a multi-task fusion architecture can learn to trust each model's representations dynamically.

Looking ahead, we advocate a targeted annotation and augmentation pipeline that focuses scarce human effort where it matters most. By combining linguistically informed strategies with model fusion strategies, we anticipate substantial gains in POS-tagging robustness for MC, and we hope this work serves as a blueprint for other under-resourced, contact-influenced languages.

Limitations

This study is constrained by the relatively small size of the annotated MC corpus, which limits model generalization and makes evaluation sensitive to lexical overlap. Additionally, while we

provide detailed post hoc analysis, our tagging architectures remain model-centric.

Future work will address these gaps by 1) Developing a reward-modeling framework to guide annotation and tagging across ambiguous categories; 2) Exploring ensemble and voting-based approaches informed by per-tag performance; 3) Designing linguistically controlled data augmentation pipelines, including LLM-generated MC sentences; and 4) integrating POS tagging with downstream tasks such as dependency parsing and translation, forming part of a unified Creole NLP pipeline.

Together, these directions aim to move Creole NLP beyond passive transfer and toward linguistically-aware, low-resource-first modeling strategies.

References

Jean Bernabé. 1983. *Fondal-natal: grammaire basilectale approchée des créoles guadeloupéen et martiniquais: approche sociolittéraire, sociolinguistique et syntaxique*. L'Harmattan.

Alebachew Chiche and Betselot Yitagesu. 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1):10.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Marcos Garcia and Pablo Gamallo. 2010. Using morphosyntactic post-processing to improve pos-tagging accuracy. In *Proceedings of the International Conference on Computational Processing of the Portuguese Language (PROPOR'2010). Porto Alegre, RS*.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*.

Zar Zar Hlaing, Ye Kyaw Thu, Thepchai Supnithi, and Ponrudee Netisopakul. 2022. Improving neural machine translation with pos-tag features for low-resource language pairs. *Heliyon*, 8(8).

Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, and 1 others. 2024. A survey on large language models with multilingualism: Recent advances and new frontiers. *arXiv preprint arXiv:2405.10936*.

Heather Lent, Kushal Tatariya, Raj Dabre, Yiyi Chen, Marcell Fekete, Esther Ploeger, Li Zhou, Ruth-Ann Armstrong, Abee Eijansantos, Catriona Malau, and 1 others. 2024. Creoleval: Multilingual multitask benchmarks for creoles. *Transactions of the Association for Computational Linguistics*, 12:950–978.

Kathe Managan. 2016. The sociolinguistic situation in guadeloupe: Diglossia reconsidered. *Journal of Pidgin and Creole Languages*, 31(2):253–287.

Ludovic Mompelat, Daniel Dakota, and Sandra Kübler. 2022. How to parse a creole: When martinican creole meets french. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4397–4406.

Salikoko S Mufwene. 2013. Simplicity and complexity in creoles and pidgins: What's the metric? *Journal of Language Contact*, 6(1):161–179.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Lambert-Félix Prudent. 1981. Diglossie et interlecte. *Langages*, (61):13–38.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*.

Matthias Schöffel, Marinus Wiedner, Esteban Garces Arias, Paula Ruppert, Christian Heumann, and Matthias Aßenmacher. 2025. Modern models, medieval texts: A pos tagging study of old occitan. *arXiv preprint arXiv:2503.07827*.

Pascal Vaillant. 2023. Noun phrases in mixed martinican creole and french: Evidence for an underspecified language model. *Journal of Pidgin and Creole Languages*, 38(2):207–262.

Lu Wang, Yahui Guo, and Chengcheng Ren. 2021. A quantitative study on english polyfunctional words. *Glottometrics*, 50.

Houquan Zhou, Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Is pos tagging necessary or even helpful for neural dependency parsing? In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 179–191. Springer.

EventHopNLI: A Functional Dataset for Systematically Diagnosing Logical Failures in LLM Temporal Reasoning

Ved Mathai
University of Oxford, UK
ved.mathai@eng.ox.ac.uk

Janet B. Pierrehumbert
University of Oxford, UK
janet.pierrehumbert@oerc.ox.ac.uk

Abstract

This paper presents EventHopNLI, a simplified functional diagnostic dataset for the task of event temporal ordering. This paper uses this diagnostic dataset to improve the interpretability of the performance of attention-based language models on this task. Existing datasets based on natural data have multiple overlapping linguistic features. Simplifying and isolating these features improves interpretability. EventHopNLI is a programmatically-created NLI dataset that systematically varies over various complexity factors such as number of events, number of logical hops etc. Even though EventHopNLI is highly simplified, it still proves challenging to language models. Being functional, the dataset is dynamic. This reduces the risk that the data is available to language models during training. We ablate over the different complexity parameters and illustrate different shortcomings of attention-based models at this task. We discuss the performance of RoBERTa-large, Llama-405B and GPT-4o. The code and data is available at https://github.com/vedmathai/eventhopnli.

1 Introduction

Identifying events in time and reasoning about their relationships is critical for many NLP application areas such as text summarization and fact checking. However, off-the-shelf large language models perform rather poorly in temporal reasoning (Xiong et al., 2024; Wang and Zhao, 2023). Despite the advances they report in building explicit temporal graphs and applying chain-of-thought reasoning, the problem cannot be viewed as solved. One reason the task is difficult is that multiple linguistic factors need to be brought to bear concurrently. A related challenge is that human annotators find it difficult and sometimes confusing to supply explicit temporal annotations, and as a result annotated natural data is expensive and often noisy. A core challenge – and the one that the present paper addresses – is that reasoning about temporal relationships is a multi-hop reasoning problem on a graph of relationships, and classic algorithms for solving such problems exactly are recursive.

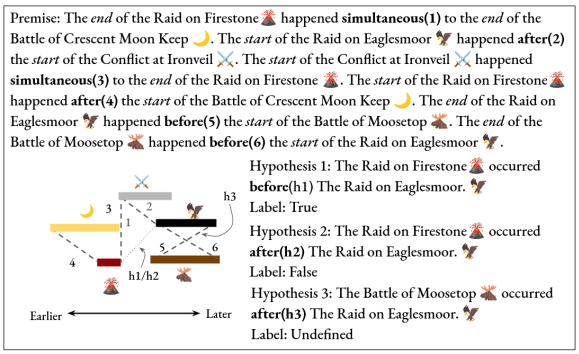


Figure 1: An example of the generated premise and an illustration of the timeline that the premise depicts. The actual dataset does not contain the emojis. They have been included in the figure to help readers parse the paragraph. The dataset is created to test the upper-bound of language models on the the temporal ordering task. The current naturalistic datasets are have confounding linguistic features making it hard to identify specific areas of improvements. Simplifying the task to this format helps us better understand models' ability to perform multi-hop reasoning for temporal reasoning.

Most annotated temporal datasets (UzZaman et al., 2013) are created by asking experts or crowdworkers to provide temporal labels on naturally occurring text articles. In general, the set of temporal relationships in a text has a $\mathcal{O}(n^2)$ complexity on the number of events. Annotating all relationships quickly becomes intractable as the text length increases, so it is necessary to select events for which relations are to be annotated. UzZaman et al. (2013) leaves the choice of the pair of events for which a relationship is to be identified to the annotators. As a result, these datasets are sparsely labeled. Later annotation efforts attempt to remedy the sparsity problems. For example, Cassidy et al. (2014) make annotators provide temporal relationship labels between all events in a two-sentence

Proceedings of the 2025 CLASP Conference on Language models And RePresentations (LARP), pages 11–27

window. Ning et al. (2018) perform two-rounds of annotations. The first classifies the member of events to multiple orthogonal axes, i.e., one axis for events that have happened and another axis for events that are only planned to happen but haven't as yet. The second round of annotation then places these events in temporal order on their respective axes. This discussion illustrates that the annotation completeness is already limited by the specific design of the annotation process.

One additional – and critical – limitation of current datasets is *Lack of diversity*. Most annotation efforts were carried out on news data, which is dominated by the past tense. This introduces biases which can limit performance on other types of text, such as planning documents. A second critical limitation is linguistic coverage. The materials are not balanced or controlled for the linguistic markers of temporal relations. This makes it impossible to carry out a diagnostic evaluation of which types of expressions are the causes of difficulties in performance.

Both of these points motivate EventHopNLI. The dataset focuses on analysing the performance of models when multiple logical hops are required to perform the temporal reasoning. We systematically vary the complexity in terms of number of events, relationships, hops and analyse the performance. An inventory of the full variety of linguistic features found in naturalistic data can be found in §2. A technical description of how the EventHopNLI dataset isolates and balances for the specific features in its scope is found in §3.

Our contributions are the following:

1. **The dataset**: EventHopNLI, which is presented in the form of a Natural Language Inference task. Each datapoint presents a paragraph (premise) and a corresponding claim (hypothesis) about the temporal order between two events in the paragraph. The task posed to the model is to decide whether the claim is true, false or undecidable due to logical inconsistencies in the text.

2. **An illustration of programmatic data generation for the task of temporal relationship extraction**: As discussed above, EventHopNLI is designed to alleviate the problems of imbalance and excessive overlap of linguistic phenomena by generating the data points programatically. This forms a functional test (Fan et al., 2024). Static benchmarks run the risk of being accessible to language models during their training. Evaluating on data that the language model has already seen during their training doesn't give us a true estimated of their expected behaviour on out-of-domain data. Functional tests create data that tests a particular functionality but has enough adaptation to the lexical representation of the data. Therefore, it guarantees that the evaluation provides a true estimate on the out-of-domain examples. We hope that this, along with the previously motivated argument to isolate linguistic phenomena when designing temporal datasets provides a template for future temporal datasets targeting other linguistic phenomena.

3. **Analysis of the model performance on this dataset.** We report the performance of three models on EventHopNLI. We find that even the best-performing model struggles on the dataset, for all but the simplest premises. Given that we have distilled down the dataset to a very simplified version with all the additional challenges removed, it is clear that reasoning capabilities beyond those provided by a transformer model using pure attention will be required to fully solve the problem of temporal-order classification.

The rest of the paper is as follows: §2 analyses the different linguistic features a model would need to have the ability to understand in order to perform the task of temporal-ordering. §3 describes the data and its creation. §4 describes the experiments performed on the dataset, followed by the presentation and analysis of the results in §5.

2 Desiderata

2.1 Target particular linguistic features while abstracting away from other features

Understanding temporal relationships involves understanding multiple linguistic features.

We argue that to systematically understand the failure modes of language models, datasets have to be designed such that linguistic features are isolated from each other in the datasets.

The specific linguistic feature that EventHopNLI focuses on are **temporal markers** such as *before*, *after*, *during* etc. that position events in relations to each other in time. EventHopNLI analyses how language models perform on chains of such relationships in text.

Below, we enumerate the linguistic features that often co-occur in temporal and event expressions. The text in EventHopNLI is designed such that the choice of label does not depend on any of the following features. In Appendix A, we take an exam-

ple from the TempEval dataset and show how these features often occur in overlapping ways. Note that each of the following are often studied substantially in isolation in theoretical linguistics. For each, we provide an explanation that highlights the main difficulty of that feature.

Events embedded under a speech-act verb: For example, *fly* in *John said that Mary will fly tomorrow*, is embedded under *said*. The temporal location of embedded events is understood in relation to that of the speech act.

Events presupposed by a corresponding state: Referring to an entity being *dead* presupposes that a *death* event happened sometime before.

Vagueness: Stating that World War II happened during the last century specifies a broader time window than saying that it took place from 1939 to 1945. Such vagueness can lead to uncertainty in inferences about temporal ordering.

Accomplishments/Achievements/Processes: If a process of drawing a circle stopped halfway then a circle is not drawn. But if the process of jogging is stopped halfway, it is still true that jogging took place.

Entity coreference resolution: The same real-world event may be referred to by different lexical descriptors. These co-references have to be correctly resolved to create the chain of event occurrences. A related yet important point is that a noun such as *recovery* alludes to a verbal event of an entity *recovering*.

Irrealis events are events that were or are planned to occur and haven't or are yet to occur. These are more difficult to model on a timeline because they may not occur or many simultaneous timelines of the future may exist.

General knowledge and commonsense: Inaugurating a new president of the USA entails that an election has taken place. Understanding this entailment involves the use of world knowledge about how elections occur.

In the next section, we show how the text of EventHopNLI is designed such that a model that has to solve EventHopNLI task (described in the next section) does not have to understand any of the linguistic features mentioned above.

2.2 Provide for rich ablations

Balancing the data across different parameters and labeling individual data points with the parameter value allows us to recognize patterns of failures in a model's behaviour. The data should be balanced for size and temporal relationships. The dataset should tackle questions such as: Are models able to decipher ambiguity in the facts stated? Do language models benefit from the temporal relationships being presented in a sorted order? It should also probe whether difficulties are intrinsic to the problem or to the lexical form of the temporal domain.

2.3 Avoid the need for human annotations

In §1 we discussed the general difficulty previous annotation efforts have faced since this task is difficult for humans too. By programmatically generating the data we can i) inexpensively create a large dataset ii) balance the dataset across a range of attributes and their values iii) guarantee the availability of a ground truth, correct, answer for each query. Inspired by (Kim and Schuster, 2023), we argue that it is beneficial to create data programmatically, with controllable parameters that systematically limit the number of linguistic factors that need to be simultaneously used.

Generating data does not preclude the need for annotated natural data. Performance metrics obtained on annotated natural data gives us a good understanding of how models will perform on data that is available in production. However, in order to improve their performance, it is important to diagnose where, why and how they fail so that future interventions can be made to either the algorithm or training data to improve performance. Testing on data that is isolated by linguistic features helps us understand the features that prove difficult for the model. Such data with isolated linguistic features can be obtained by filtering natural data, which is labour intensive and expensive. Alternatively, it can be obtained by generating data: the method used by this paper.

3 Data

Formally, we have a set of events $\mathcal{E}$; and relationship types $\mathcal{T}$ = {*before*, *after*, *simultaneous*}. Each event has a *start* and an *end*. The premise is a set of temporal relationships r that are in the format of a triple ($e_1^{start/end}$, t, $e_2^{start/end}$) where $e \in \mathcal{E}$ and $t \in \mathcal{T}$, the superscript of e indicates the extremity and the subscript indicates the index of e. The hypothesis is of the form (e_1, r, e_2) where $r \in \mathcal{R}$ and $\mathcal{R}$={*before*, *after*, *overlaps*}. The label $l \in \mathcal{L}$ and $\mathcal{L}$ = {*true*, *false*, *undefined*}. The task is a modification of a natural language inference (NLI) task.

Each data point consists of a premise, a hypothesis (or claim) and a label.

The premise is a set of n relationships between m events while the hypothesis makes a claim about a single relationship r. A simpler formulation of the problem would involve the temporal relationship between point events. However, we choose to test durative events because they form an inherently harder problem and one that forms a better representation of those events available in the natural texts. In our formulation, one has to keep track of the extremities of the event and compare all four points in order to find the ordering of events versus just the two required of point events. This includes keeping track of when the events mentions are under-specified, i.e., only the start or the end of an event is mentioned.

1) ***true*** labels cases where the hypothesis is true given the premise. 2) ***false*** labels cases where the hypothesis is false given the premise. A pair can be false either because an alternative claim is true or no claim can be derived. For example: if the premise makes a claim (e_1^{start}, before, e_2^{end}) and the hypothesis makes the claim (e_1^{start}, after, e_2^{end}). Then the pair is labeled as *false*. 3) ***undefined*** labels cases when there are two more logical paths that involve the two events that present in a cyclical chain. For example, i.e., both (e_1^{start}, before, e_2^{end}) and (e_1^{start}, after, e_2^{end}) exist or are derivable from the evidence available.

In general, there are 13 different temporal relationships (Allen, 1983). It is undesirable to tie the architecture of a model to the subset of relationships that need to be classified. Setting up an NLI task allows the set of labels to remain constant while the set of temporal relationships tested can be varied easily without affecting existing learning architectures.

Examples of generated data points are presented in Fig.1. The following are particular design choices and the particular desiderata that they address:

Dataset is agnostic to world knowledge: Our dataset has the flavour of a miniature language that could be employed in the context of a multi-player on-line game. By using fictional battle names such 'The Raid on Firestone', we guarantee that the proposed evaluation methodology is testing the model's ability to understand the logic of temporal relationship markers and not using any facts about real historical battles that it may have learned during its pre-training phase.

Minimalistic descriptions of temporal relationships: The premise is made up of only relationships in the form of start/end of event 1 after/before/simultaneous start/end of event 2. Using the full names of events in the specification of every relationship removes the need for models to apply entity coreference resolution that is more complex than simple lexical matching. No commonsense or world-knowledge beyond that of the temporal relationship markers has to be applied. There is no verbal event therefore there is no need to understand tense. All events are realis, (i.e, have actually occurred versus possibly occurring in the future) and are named events (i.e, there is no use of anaphora or events rooted in verbs). There are no descriptions of states.

Ablating over sizes: Ablating over the sizes of the premise in terms of relationships, events, logical hops[1] gives us an understanding of how increased complexity due to size and increased number of relationships affect language models' performance.

By ablating over the **temporal relationships** in the premise we can identify if any particular temporal relationship proves to be more difficult than others. We predict that *simultaneous* would be more difficult than that of *before* and *after* for the following reasons.

Finding the temporal order between events or event identifying contradictions in the premise involves performing a depth first search on the graph. Traversing a graph with only *after* and *before* relationships is computationally easier than traversing a graph with *simultaneous* relationships. Both traversals involve maintaining a stack. However, with *simultaneous* relationships the stack size at any given point can be larger.

We explain with an example: Assume event *A* is earlier than event *B*. It is not necessary to add events earlier than *A* or later than B to the stack since they do not provide useful information to the temporal chain that connects event *A* and event *B*. However, assume a simultaneous relationship between *A* and *C*. Now all of the nodes related to *C* will have to be added to the stack, which means that with many simultaneous relationships there are

[1]We define a **temporal chain** to be a sequence of two or more events ordered temporally such that traversing the chain provides the temporal relationship between the two events at its endpoints. We further define each individual relationship on the chain to be a logical hop.

higher chances of the size of the stack being larger than if there were no simultaneous relationships.

Sorting: Providing the temporal relationships in the premise in a sorted order resembles how events are often described in text, i.e., in temporal order. Sorting relationships reduces the number of permutations in which the same timeline can be expressed to just one. This would increase the probability that patterns between the train and test sets repeat.

Comparison with other domains: Testing on only the temporal domain raises the question of whether the difficulty of the task is intrinsic to the task of timeline resolution or to the lexical properties of the temporal relationships. The dataset consists of computationally equivalent tasks but in two following domains:

i) **Spatial domain**: Events are mapped to locations and the start and stop points are mapped to east and west edges and the temporal relationships are mapped to *east of* and *west of*. An example is provided in Table 4.

ii) **Logical axioms:** Relationships are mapped to $\{<, >, =\}$ and the event names to simple tokens. This formulation strips away the natural language aspect of the task.

Generalizability versus memoization: We expect a language model to generalize the logic of temporal relationships from the training data while not memoizing (learning by rote) event names, specific lexical forms or timelines. We create train-test pairs that individually keep event names, lexical forms or timelines common between the train and test datasets.

Impossible logical chains: A logically coherent temporal graph is a directed acyclic graph. If a cycle exists in a temporal graph then the temporal relationships among the events on the cycle are undefined. Annotated natural text data may have cyclic relations either because the text itself contains contradictions, or because the annotators made mistakes. A language model should recognize such cycles and report them. Our dataset contains instances of cycles only by design, when the claims in the text are contradictory.

A note on complexity of the task: Since the data points were created using an algorithm and natural language templates, it is trivial to write a parser that parses the data points back into a temporal graph to obtain 100% accuracy. This places performance of the language models in perspective. Kim and Schuster (2023) argues that their task of entity-tracking may be hard for humans when the stimuli, which are quite long, are presented orally. However, when provided with a scratchpad, humans were able to solve the task exactly. Our task is similar in that it can be solved exactly with a scratchpad and careful reasoning. Human annotators may not obtain a perfect score due to factors such as fatigue, carelessness or simple errors. However, when provided with a written version, a scratchpad and no time constraints, one of the authors was able to solve the question exactly. With these experiments, we provide similar standards to the language model. We ask if the language models can solve the task exactly if they had no time constraints. Language models are increasingly being deployed in applications that help students and researchers who expect a high level of correctness from the models especially on such logical problems that don't involve subjective decisions (Kooli, 2023), therefore it is instructive to understand the upper-limit of their logical performance.

3.1 Generation of the EventHopNLI dataset

The following are salient points about the implementation details of the program that generates the dataset, more information including pseudocode is included in Appendix 8.

The program has four sections: i) timeline data structure; ii) timeline generator; iii) verifier; iv) data-structure-to-natural-language translator.

Timeline data structure: The timeline itself is a graph of events connected by temporal relationships. The events are partially ordered and the dataset is balanced to have timelines with internal contradictions.

Timeline generator: The set of fictional battle names are generated using ChatGPT. We create a list of names for the test set mutually exclusive from the names in the train dataset. The extremities (start or end) of two events are selected at random and a relationship with a specific temporal relationship is generated between them.

Verifier: In some cases in the timeline graph there may be internal logical contradictions. A hypothesis is generated by choosing two events and assigning a relationship between them. The verifier traverses the generated timeline graph and decides whether the hypothesis is *true*, *false* or *undefined*.

It is possible for the relationship between the two

events in the hypothesis to be under-specified. For example, given two events A and B, the premise specifies that the start of event *A* is before the start of *B*, however, the premise does not specify if the relationship is between the end points. Assume the claim in the hypothesis is: event *A* being before, after or overlapping with *B*. It is unclear which one of these temporal orderings is true without knowing the exact temporal ordering between the end-points. Therefore we label the pair as *false*.

A closed cycle in the graph is evidence of a logical contradiction. Using a depth-first search traversal on a directed graph allows us to find cycles easily, because visiting a previously visited node would present a cycle.

Data-structure-to-natural-language translator: The temporal graph is converted to natural language descriptions using templates. Later, we describe how we create different sets of these templates for an ablation study.

3.2 Balancing EventHopNLI and defining its variations

We balance EventHopNLI across the following attributes: {number of events, number of relationships, number of hops, whether an internal logical contradiction exits, temporal relationship, label}.

The **number of events** vary over a exponentially increasing set of sizes, i.e., $\in \{4, 8, 16, 32\}$.

The **number of relationships** vary as function of the number of events. *number of relationships = min(max(number of events, 3), 32)* where *relationship_number = relationship_multiplier*$\times$ *number of events* and *relationship_multiplier* $\in \{0.5, 1, 2\}$.

The dataset represents a Cartesian product of the following form: *number of events* $\times$ *number of relationships* $\times$ *temporal relationship* $\times$ *whether there exists an internal contradiction* $\times$ *label*.

We create different train-set tests with the following strategies. Each of the strategies inform us about the models' capabilities.

Each of the train datasets are about 14,000 data points. The test set are 2,500 data points.

Strict: In this pairing, the names of events, natural language templates for the relationships and the timelines are mutually exclusive between the train and test set. Naturalistic data would have a high degree of mutual exclusivity between the train and test set, therefore an efficient model would be expected to not perform memoization.

Same names/templates/timelines: To systematically check for the models ability to generalize vs. memoizing (cf §3) we remove the mutual exclusivity between the train and test set and create the following three sets of train datasets. In each one of i) **event names** ii) the **natural language templates** used to create the premise iii) the set of **timeline graphs** are maintained as common information between the train and test set.

Random	0.323
RoBERTa standard	0.762
RoBERTa spatial domain	0.66
RoBERTa logical domain	0.79
Llama 405B strict	0.4
GPT-4o strict	0.36

Table 1: Main results reported on the standard specification of the dataset

Ablating temporal relationship type and sorting: We create the following deviations from the strict dataset by varying the relationship types allowed and sorting the relationships in the premise temporally:

i) only after and before relationships

ii) only after and simultaneous relationships

iii) only before and simultaneous relationships

iv) only before relationships: this is the same as (iii) but the *after* relationships are inverted to make them *before* relationships

v) All relationships sorted [2]

vi) Only before relationships sorted: same as (iv) but sorted temporally.

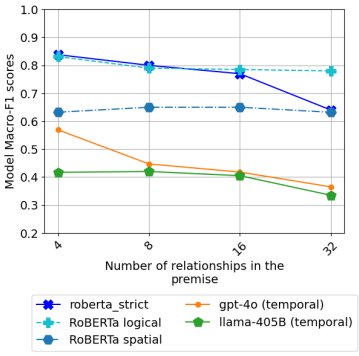
These ablations will inform us i) whether the *simultaneous* relationship-type is indeed more difficult than *after/before* ii) Whether sorting relationships temporally benefits learning because it reduces the entropy in terms of presentation of information.

Alternate domains: Following §3, we create two sets of test-train pairs. One for the spatial domain and the second for the logical domain.

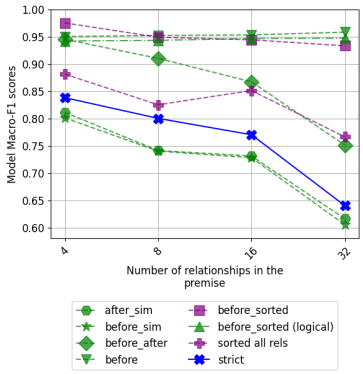
4 Experiments

Finetuning-based experiments: We experiment with fine-tuning a RoBERTa-large (Liu et al., 2019) model on the given data to see how well the model can learn this task with fine-tuning. We deliberately

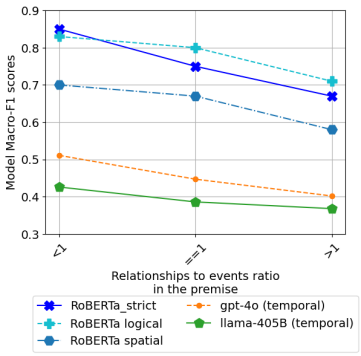
[2]All of the cases in which the premise is sorted, the relationships are sorted temporally by the earliest event present in the relationship. Arbitarily choosing events in the case of logical cycles.



(a) Performance of RoBERTa on different domains



(b) Ablation performed with RoBERTa on datasets systematically designed to maintain common information between the train and test set.



(c) Ablation study performed on the datasets with the data points partitioned by the relationship-to-event ratios in the premise.

Figure 2

use RoBERTa because it is an example of a large model which can easily be fully fine-tuned without the use of adapters.

We use a learning rate of 1e-6.

In-context demonstration: In this setting, the large language model is prompted with only the task description, label description and the expected response format as shown in Table 3. We prompt GPT-4o (Hurst et al., 2024), a large commercial closed model [3] and Llama-405b (Grattafiori et al., 2024), a large open-weight model.

It is instructive to evaluate the ability of these large-language on this task without any fine-tuning, with the understanding that a low performance on this task will automatically translate to a low performance on an even more complex task.

5 Results

We use macro-F1 as the basic metric for all of the results reported.

5.1 Simple Baselines

We report the **simple baseline** of random selection to show that the dataset is both not trivial and is balanced. An F1 score of 0.33 on the random selection shows that the dataset is balanced.

5.2 Main Results

Table 1 reports the main results. Each of the result reports the mean of five runs.

[3]We attempted experiments on GPT-o1, OpenAI's latest reasoning model and we found the the results varied drastically between multiple runs over multiple weeks. Therefore, we do not report the performance.

Performance drops as complexity increases Fig. 2a shows that as the number of relationships increases, the performance of the models deteriorates. This is an indicator of the expected performance of models on production data. Timeline creation is a direct downstream task of temporal relationship extraction and regardless of having a larger context window, lower performance can be expected from the models when the number of relationships that need to be parsed increase.

The rate of decrease in performance as the number of events increases (shown in Fig. 3a) is not as prominent as when the number of relationships increase, this indicates that it is the number of relationships and not the number of events is the source of complexity for the task.

Fig. 2c provides further evidence. It plots performance as a function of the relationships to events ratio. Decreasing performance as the ratio increases shows that the performance is more influenced by the number of relationships rather than events.

As the number of logical hops required increases, the performance first falls and then asymptotes out. (Fig. 3b). We attribute this to the fact that as the number of permutations in which smaller temporal chains are presented is exponentially smaller than the ways relationships with larger temporal chains can be presented. This makes it easier for the model to memoise the different patterns that represent the temporal chains.

Surprisingly, this effect is not seen in GPT-4o, suggesting that its higher number of parameters, larger training data and larger context window enable it to perform the same regardless of the number

of hops required to solve the particular data point, even though its overall performance is low.

Larger parameter size and pre-training data does not provide an advantage The performance of models with a larger parameter capacity on this simplified dataset shows that even in its highly-simplified formulation temporal relationship extraction is still computationally challenging for language models.

The fine-tuned RoBERTa model performs much better than the few-shot prompted larger models (Llama-405b and GPT-4o). This suggests that the data that very large language models learn from is too noisy for them to effectively learn temporal reasoning.

The complexity is intrinsic to the problem and does not arise from the lexical constructions.

In Fig. 2a, we see that the temporal domain and logical domain perform similarly, except for the case of 32 relationships, where the logical domain maintains its performance. This may be because the number of tokens is lower in the logical domain, and never exceeds the limit of the context window.

The spatial domain is intrinsically harder to parse than that of the temporal domain, which shows that the temporal domain is not a lower-limit for the task.

RoBERTa performs detrimental memoization The performance of `same_timelines` and `same_names` are both lower than `RoBERTa_strict`. This shows that during fine-tuning RoBERTa is performing memoization (i.e., learning axioms by rote) rather than learning the patterns in a generalizable manner. (Fig. 3c). This is undesirable behaviour from a model that is expected to generalize to new information.

We see in the same figure that when the test set uses the same natural language templates as the train set, performance improves which means the fine-tuned model struggles to extend to different sentence constructions that express relationships.

Sorting the relationships temporally improves the performance of the NLI classification. By sorting the timelines, the number of permutations to describe the same timeline reduces to just one. This in turn increases the probability that patterns will repeat between the train and test sets.

These results allows us to predict the performance of models of different types of data that may be encountered in production: better performance can be expected on data that follows a chronological structure than those that don't.

The simultaneous relationship greatly increases the complexity of the problem. In Fig. 2b we see that the datasets *only after and simultaneous* and *only before and simultaneous* obtain similar scores. However, *only before and after* obtains a much better score than that of the standard specification.

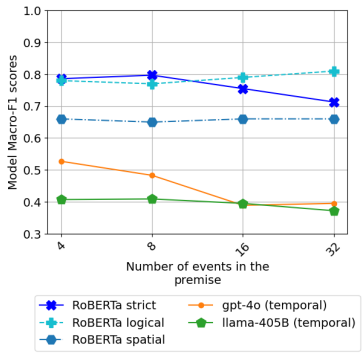
This result corroborates our intuition (cf. §3) that the formulation of the problem that included simultaneous relationships were computationally harder than those that didn't have a simultaneous relationship.

The formulation with *only before* is near ceiling Finetuned RoBERTa achieves 0.95 macro-F1 on the dataset which has simultaneous relationships removed and the *after* relationships to be inverted to be *before* relationships.

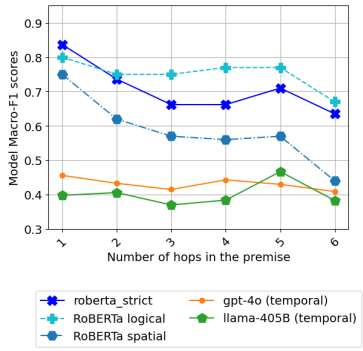
GPT is confused by the undefined class. RoBERTa is less so. Fig. 4a and Fig. 4b plots the confusion matrices for both GPT-4o and RoBERTa. Fig. 4a shows that GPT is prone to choosing the *True* or *False* labels while being averse to choosing the *Undefined* label. This shows a limitation in identifying when the premise contradicts itself. RoBERTa's confusion matrix (Fig. 4b) appears balanced in comparison. This means that an attention model is able to learn (by fine-tuning) generalizable patterns when given enough examples.

6 Related Work

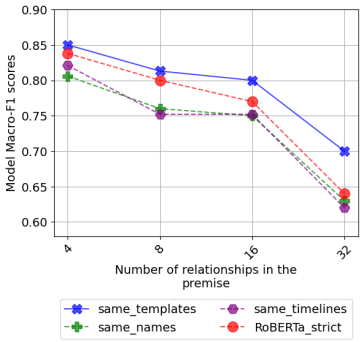
Recent studies on the performance of LLMs on temporal data (Xiong et al., 2024; Wang and Zhao, 2023) show that the problem is far from solved on test suites such as TimeBench (Chu et al., 2024). Both (Wang et al., 2024b) and (Xiong et al., 2024) attempt to improve performance by using temporal graphs. Xiong et al. (2024) show that converting a textual description into a temporal graph and performing chain-of-thought reasoning increases performance. However, the study does not systematically vary the complexity of the reasoning required, or diagnose the error patterns. We provide a systematic set of experiments and analysis in understanding the failure modes of the language models. We show that even the simplest descriptions of events are not fully understood by language models. Holtermann et al. (2025) find that models are able to satisfactorily perform temporal reasoning over timezones, they are not able to perform the same reasoning when asked to reason over both



(a) Ablation on event count performed with different models

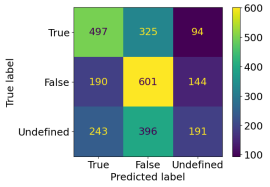


(b) Ablation study performed on the datasets partitioned by the number of logical hops in the premise (cf 2).

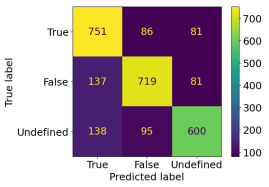


(c) Ablation performed with RoBERTa on datasets systematically designed to maintain common information between the train and test set.

Figure 3



(a) Confusion matrix on the labels for GPT-4o



(b) Confusion matrix on the labels for RoBERTa-large

Figure 4: Confusion Matrices

timezone and geographical locations.

Our work focuses on multi-hop reasoning on the specific domain of temporal graphs. Other projects (Lin et al., 2018; Misra et al., 2023) explore multi-hop reasoning on knowledge graphs. Wang et al. (2024a); Lin et al. (2024) study LLM capabilities in reasoning about graph datastructures that arise in other application areas. Their results resemble ours in showing deterioration of performance with increases in graph complexity. Works such as Han et al. (2024), explore the language models' ability to perform multi-hop reasoning on datasets that involve first-order logic (FOL) while Chen et al. (2020) explores multi-hop reasoning on textual information for the task of question answering.

Qi et al. (2024) explores how well language models are able to solve problems of different theoretical computational complexity. They show how performance degrades as the complexity increases. While the differences in complexities in their examples are more marked, the differences in complexity for temporal reasoning, as we have shown, are softer.

7 Conclusion

This paper presents EventHopNLI, a dataset that systematically identifies the error patterns of large language models on the task of temporal order. The dataset is designed to perform ablations across multiple factors (such as size of texts, temporal relationships, number of logical hops) while isolating specific linguistic features. We use the dataset to diagnose the error modes on examples of two paradigms of language models.

The results show us that there are limitations to language models' ability to traverse temporal graphs represented in texts. This prompts future research to investigate whether the use of a logical theorem solver (Pan et al., 2023; Olausson et al., 2023) can help obtain better results on the temporal ordering task.

Hopefully, this study will help everyday users of such models understand the expected limitations when they are applied to their specific data.

8 Limitations

As a result of our design goals, the dataset is limited to a specific set of linguistic features and temporal relations. It does not cover the further features described in §1 and §2. We leave it to future studies to create equally controlled diagnostic datasets that systematically include more of the linguistic features described in §2 such that the gap between natural data and this synthetic data is closed. We only evaluate general-purpose large language models, and do not evaluate approaches that explicitly construct temporal graphs or use scratchpads or Chain-of-Thought reasoning.

Acknowledgments

This work was funded by the Engineering and Physical Sciences Research Council (grant EP/T023333/1 awarded to University of Oxford)

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, Bangkok, Thailand. Association for Computational Linguistics.

Lizhou Fan, Wenyue Hua, Lingyao Li, Haoyang Ling, and Yongfeng Zhang. 2024. NPHardEval: Dynamic benchmark on reasoning ability of large language models via complexity classes. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4092–4114, Bangkok, Thailand. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Carolin Holtermann, Paul Röttger, and Anne Lauscher. 2025. Around the world in 24 hours: Probing llm knowledge of time and place. *arXiv preprint arXiv:2506.03984*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855.

Chokri Kooli. 2023. Chatbots in education and research: A critical examination of ethical implications and solutions. *Sustainability*, 15(7).

Fangru Lin, Emanuele La Malfa, Valentin Hofmann, Elle Michelle Yang, Anthony G Cohn, and Janet B Pierrehumbert. 2024. Graph-enhanced large language models in asynchronous plan reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 30108–30134.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kanishka Misra, Cicero Nogueira dos Santos, and Siamak Shakeri. 2023. Triggering multi-hop reasoning for question answering in language models using soft prompts and random walks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 972–985, Toronto, Canada. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the*

2023 Conference on Empirical Methods in Natural Language Processing, pages 5153–5176, Singapore. Association for Computational Linguistics.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Zhenting Qi, Hongyin Luo, Xuliang Huang, Zhuokai Zhao, Yibo Jiang, Xiangjun Fan, Himabindu Lakkaraju, and James Glass. 2024. Quantifying generalization complexity for large language models.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. 2024a. InstructGraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13492–13510, Bangkok, Thailand. Association for Computational Linguistics.

Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2024b. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pages 8384–8410. Curran Associates, Inc.

Yuqing Wang and Yun Zhao. 2023. Tram: Benchmarking temporal reasoning for large language models. *arXiv preprint arXiv:2310.00835*.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10452–10470.

A Analysis of the general complexity of temporal relationship extraction

In this section, we analyse a single paragraph from the TempEval dataset and illustrate the general complexity of the temporal relationship extraction task, both for models and for the human-annotation efforts. We identify and list below linguistic features or structures that provide the complexity.

Embedded under a speech-act verb: The event *sent* at (3) is embedded under the speech-act verb (1). The time of occurrence of the (3) is dependent on the time of occurrence of (1).

Expression of states indicate events that caused them: (5) indicates a current state of the expert, i.e., dead. This state indicates that a death had occurred sometime before. The *death* event remains implicit in the paragraph.

Vagueness: Compare temporal expressions (6) and event (14). The modifier *almost* introduces vagueness to the temporal expression. (14) is a temporal expression signifying a particular particular month.

Accomplishments/Achievements/Processes: The *recovery* at (7) is an accomplishment which culminates a period of searching which in itself is an implicit durative process. This implicit link will have to be processed by annotators and models.

Irrealis events: (7) is an infinitive subordinate of 'hope'. This places it on the irrealis axis. This means that though the event is reported it is unknown if the event definitely happened if it is in the past, or if is going to happen for sure if it is in the future. Many annotation schemes ask annotators to ignore such events since it is hard to provide a label for events that may take place. Similar problems exist for negative events.

Entity coreference resolution: (4) refers to a person (an entity) by their occupation while (8) refers to the entity by their name, without drawing an explicit connection between the two references. Entity coreference is itself a complex unsolved problem in NLP. Its complexity is inherited in the general task of temporal relationship classification.

Temporal markers: (9), (12) and (13) marks an explicit temporal relationship between an event and time. Obviously, having explicit temporal markers between events greatly simplifies the problem as compared to using implicit knowledge; however, an aim of this paper is to stress-test models' understanding given materials with only these explicit markers.

The top commander of a Cambodian resistance force said[1] Thursday[2] he has sent[3] a team to recover the remains of a British mine removal expert[4] kidnapped and presumed killed[5] by Khmer Rouge guerrillas almost two years ago[6]. Gen. Nhek Bunchhay ... said in an interview with The Associated Press at his hilltop headquarters that he hopes to recover[7] the remains of Christopher Howes[8] within[9] the next two weeks. Howes had been working[10] for the Britain-based Mines Advisory Group when[11] he was abducted[12] with his Cambodian interpreter Houn Hourth in[13] March 1996[14].

Table 2: An abridged example of text and events from the TempEval dataset. A popular dataset for evaluating temporal relationship extraction. This paragraph has been taken from article APW19980219.0476.tml. Each of the linguistic features and a discussion for them is presented in §2.

General knowledge and commonsense: (12) *working* can only happen before (7) because a person can only be working if they are alive. This is an example of general knowledge and commonsense being applied.

Therefore the task of temporal relationship extraction involves the application of multiple linguistic faculty. These features interact deeply with each other further complicating the task.

Algorithm 1 Find if event_1 overlaps event_2: Check if ep^1_{start} before ep^2_{start} and ep^1_{end} is after ep^2_{start}. Note that this assumes that e1 is before e2. The function will have to be called twice with swapped parameters to check all conditions of overlap.

```
function DOES_OVERLAP_FORWARDS(e1, e2)
    s1_s2 ← is_ep1_before_ep2(e1.sp, e2.sp)
    s2_s1 ← is_ep1_before_ep2(e2.sp, e1.sp)
    e1_s1 ← is_ep1_before_ep2(e1.ep, e1.sp)
    s2_e1 ← is_ep1_before_ep2(e2.sp, e1.ep)
    e2_s2 ← is_ep1_before_ep2(e2.ep, e2.sp)
    check ← (NOT e1_s1) AND (NOT e2_s2) AND ((NOT s2_s1) OR s1_s2) AND s2_e1
return check

end function
```

Algorithm 2 Find if event_point_1 occurs before but not simultaneous to event_point_2: Follow the relations and adding events to the queue if they happen after or simultaneous to those already in the queue

```
function IS_EP1_BEFORE_EP2(ep1, ep2)
    eps ← [(ep1, True)]
    seen ← set()
    while eps.length > 0 do
        ep, is_sim ← eps.pop()
        if ep in seen then
            continue
        end if
        seen.add(ep)
        for rel in ep.rels do
            if rel.reltype == 'after' AND ep == rel.ep2 then
                eps.append((rel.ep1, False))
            end if
            if rel.reltype() == 'before' AND ep == rel.ep1 then
                eps.append((rel.ep2(), False))
            end if
            if rel.reltype == 'simultaneous' then
                eps.append((rel.other_point(ep), is_sim))
            end if
            if (ep2, False) in eps then return True
            end if
        end for
    end while
     return False
end function
```

Algorithm 3 Find if the relationship between event_1 and event_2 cannot be determined because there is contradictory evidence. Check if there is contradictory evidence between all pairs of extremities

```
function IS_CONTRADICTORY_EVENT_PAIR(e1, e2)
    s1_s2 ← is_contra_eps(e1.sp, e2.sp)
    s1_e1 ← is_contra_eps(e1.sp, e1.ep)
    s1_e2 ← is_contra_eps(e1.sp, e2.ep)
    s2_e2 ← is_contra_eps(e2.sp, e2.ep)
    check ← s1_s2 or s1_e1 or s1_e2 or s2_e2
return check

end function
```

Algorithm 4 Find if event_point_1 and event_point_2 have contradictory relationships: Check if both ep1 is before and after ep2 as long as they are not simultaneous. If they are of the same event then make sure the end is not before start

```
function IS_CONTRADICTORY_EVENT_POINTS(ep1, ep2)
    same_event_check ← (ep2.event == ep2.event AND (ep1.event.stp == ep1) AND
            (ep2.event.enp == ep2)
    check_forwards ← is_ep1_before_ep2(ep1, ep2)
    check_backwards ← is_ep1_before_ep2(ep2, ep1)
    check_simultaneous ← is_simul_eps(ep1, ep2)
    check ← check_forwards AND check_backwards AND NOT check_simultaneous and
            NOT (same_event_check and check_backwards))
     return check

end function
```

Algorithm 5 Find if event_1 overlaps with event_2: Check if the events are not contradictory and if they overlap.

```
function IS_OVERLAP_EVENTS(e1, e2)
    is_contradictory ← is_contradictory_event_pair(e1, e2)
    e1_e2 ← self.does_overlap_forwards(e1, e2)
    e2_e1 ← self.does_overlap_forwards(e2, e1)
    check ← NOT is_contradictory AND (e1_e2 OR e2_e1)
     return check
end function
```

Algorithm 6 Check if ep1 and ep2 are simultaneous. Accumulate events in a queue by adding event points to the queue that are simultaneous to those already in the queue.

```
function IS_SIMUL_EPS(ep1, ep2)
    eps ← [event_point1]
    seen ← set()
    while eps.length > 0 do
        ep = eps.pop()
        seen.add(ep)
        for rel in ep.rels do
            if rel.reltype == 'simultaneous' then
                other_point ← rel.other_point(ep)
                if other_point NOT in seen then eps.append(other_point)
                end if
            end if
            if ep2 in eps then return True
            end if
        end for
    end whilereturn False
end function
```

Algorithm 7 Find if event_1 occurs strictly before event_point_2: check if ep^1_{start} and e^1_{end} are both before ep^2_{start} and e^1_{end} is not simultaneous to e^1_{end}

```
function IS_STRICTLY_BEFORE(e1, e2)
    s1_s2 ← is_ep1_before_ep2(e1.startp, e2.startp)
    e1_s2 ← is_ep1_before_ep2(e1.endp, e2.startp)
    e1_s2_is_simul ← is_simul_eps(e1.endp, e2.startp)
    check ← s1_s2 AND (e1_s2 OR e1_s2_is_simul)
     return check
end function
```

[INST] «SYS»
The premise is a set of battles and their temporal relationships
The hypothesis is a claim of the temporal relationship between two battles.

There are three answer choices:
1) True: The hypothesis is true given the premise
2) False: The hypothesis is False given the premise
3) Undefined: There is logically contradictory evidence in the premise regarding the events in the hypothesis. So no claim can be made.

The first five are examples with the labels provided.

Your task is to predict the label for the given examples. Do not provide reasoning and provide in the format of 'answer: index: label'.

Examples: <examples>

«/SYS»

Provide the labels for the following sentences in the format of 'answer: index: label'.

<uid> premise: <premise>
hypothesis: <hypothesis>

[INST]

Table 3: The baseline prompt used for Llama. The tokens in <> are replaced by actual values from the dataset.

Western edge of Stormforge is located to the east of eastern edge of Bloodmoon Keep. Eastern edge of Stormforge is located to the east of eastern edge of Bloodmoon Keep. Western edge of Sunfire Canyon is located to the west of western edge of Bloodmoon Keep. Eastern edge of Frostfang Pass is located to the west of western edge of Ravenloft. Eastern edge of Bloodmoon Keep is located to the west of eastern edge of Ravenloft. Eastern edge of Frostfang Pass is located to the east of western edge of Sunfire Canyon. Western edge of Sunfire Canyon is on the same longitude as western edge of Frostfang Pass. Eastern edge of Bloodmoon Keep is on the same longitude as western edge of Ravenloft. Western edge of Stormforge is located to the east of eastern edge of Ravenloft.

Table 4: Example of the NLI data in the spatial domain.

Combining Information State Update, Harel Statecharts and LLMs for controllable and flexible Conversational AI

Vladislav Maraev[*,†] **and Alexander Berman**[*] **and Staffan Larsson**[*,†]
[*]Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg, Sweden
[†]Talkamatic AB, Sweden
Correspondence: vladislav.maraev@gu.se

Abstract

The rise of LLM-based approaches to dialogue systems has created an increased need for controllable dialogue. This paper addresses this need by presenting an implementation of a dialogue system based on information state update approach according to Larsson (2002). This enables the integration of rule-based handling of dialogue, expressed by Harel's statecharts (1987), and Larsson's theoretical account grounded in theories of dialogue, expressed by information state update rules. We demonstrate how our approach applies to dialogue domains involving form-filling. We also propose how LLMs can be employed to inject domain knowledge and be used in various components of a hybrid dialogue system, while maintaining control over the overall dialogue logic.

1 Introduction

Despite considerable efforts to control large language models (LLMs), risks of hallucinations and related phenomena have yet to be eliminated (Xu et al., 2024; Ayyamperumal and Ge, 2024). Still, the otherwise impressive capabilities of LLMs have raised the bar for conversational AI. This makes a compelling argument for finding ways to complement LLMs with rule-based approaches that can mitigate the risks associated with using LLMs.

In this work we use the influential Information State Update (ISU) framework (Larsson and Traum, 2000; Larsson, 2002). The basis of this framework is a representation of the dialogue context as data structure which includes the information available to each participant of the dialogue (either a human or an artificial agent). Being rich entails that the information state contains a hierarchy of facts, including the ones that are thought to be shared and the ones that have not been yet publicised. We believe that ISU is a good basis for formalising existing theories of human-human dialogue (Cann et al., 2005; Ginzburg, 2012; Cooper, 2022) as a means to develop dialogue systems which produce acceptable and natural behaviours.

In controlled ISU-based systems, LLMs offer benefits such as interpreting user utterances and pre-generating dialogue domains (Larsson, 2024). Pre-generating a complete dialogue description (all possible dialogue paths) is a complex matter due to the inherent flexibility and variation in natural language dialogue. By taking care of a lot of this complexity using a flexible but controllable dialogue manager, the ISU approach considerably simplifies the pre-generation process. Instead of generating all possible dialogue paths, it is sufficient to generate a default dialogue "blueprint" which can then be used by rule-based dialogue manager as a resource when carrying out flexible and complex dialogue interactions.

At the same time, not all dialogue behaviours are necessarily best treated on an ISU level. This applies in particular to more routinised and "reflex" behaviours such as those involved in real-time incremental turntaking (Skantze, 2021; Howes et al., 2019, among others). Such behaviours are arguably better coded as statecharts.

So, how do we best combine ISU-based dialogue management, statecharts, and LLMs for flexible and controllable AI? We argue that this is best done by offering a single framework encompassing both ISU and statecharts for managing dialogue based on LLM-generated dialogue blueprints. We term our approach Statecharts-based implementation of Information State Update (SISU).[1]

The paper is organised as follows. Section 2 introduces the main considerations behind SISU. In Section 3 we describe the implementation of SISU, while Sections 4 and 5 provide an outlook on two main advantages of our framework: pre-generating

[1]Source code and online demo available at: https://github.com/GU-CLASP/sisu

Proceedings of the 2025 CLASP Conference on Language models And RePresentations (LARP), pages 28–37

dialogue blueprints and coding low-level dialogue routines with statecharts. We provide brief conclusions in Section 6.

2 Method

This section introduces SISU, an ISU framework which combines the statecharts formalism (Harel, 1987) with modern features of the TypeScript programming language. SISU implements a version of IBiS1 (Larsson, 2002) and improves upon it by providing the capacity to code routinised dialogue phenomena using statecharts.

2.1 Information State Update

One of the central purposes of the ISU approach to dialogue management (Larsson and Traum, 2000; Larsson, 2002) is to enable the implementation and comparison of dialogue theories by casting them in a common form. Central questions in this endeavour are: (1) what kind of information does a dialogue participant need to keep track of, (2) how does this information get updated by utterances in dialogue and (3) how does this information license subsequent utterances?

In the ISU approach, utterances in conversation are seen as *dialogue moves* which trigger updates to a rich conversational Information State (IS), which then is used to select an appropriate followup dialogue moves. *Update rules* describe updates to the IS, and *selection rules* describe conditions on IS under which a move can be selected. For example, an information state can include a stack of questions that have been raised but not yet addressed. Given this, one update rule may state that asking a question Q results in pushing Q on the question stack, and a selection rule may state that if a question is on the question stack, and if an answer A to that question is known, an answer move with content A may be selected and uttered in the next turn.

An example of an ISU-based dialogue manager is Talkamatic Dialogue Manager (TDM) (Larsson and Berman, 2016) which is a core component of Talkamatic Studio[2], a web-based tool for generating, curating and deploying Pre-Generative Conversational AI agents. TDM handles a wide variety of dialogue behaviours and a wide range of dialogue types, including question-answering, search, device control, educational, instructional and negotiative dialogue. TDM uses Issue-Based Dialogue Management (Larsson, 2002) which is strongly influenced by KoS (Ginzburg, 2012).

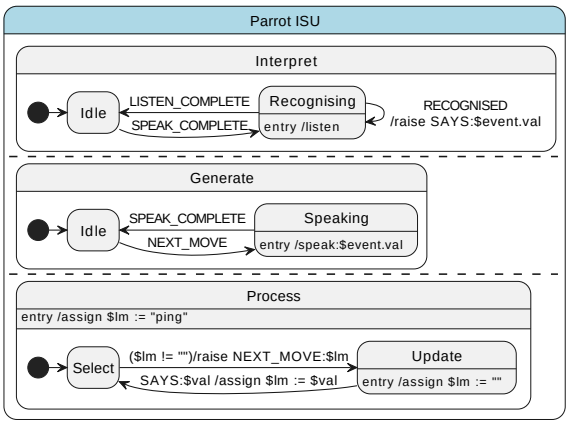


Figure 1: A statechart for "parrot" dialogue system. Events are typeset in all capitals, states – with initial capital. Conditions are parenthesised and actions are denoted by a starting "/" symbol.

2.2 Statecharts

Statecharts were originally developed by Harel (1987) for complex systems (real-time, multi-computer and concurrent). Harel introduces conventional notation over deterministic Finite-State Machines (FSMs), incorporating depth, orthogonality and broadcast communication. Statecharts can be represented graphically, as diagrams with a variable level of detail. A number of studies have demonstrated that statecharts can be useful for designing dialogue systems (Kronlid and Lager, 2007; Brusk, 2008; Mehlmann et al., 2011).

Figure 1 illustrates the use of three orthogonal (parallel) states in a simplest "parrot" dialogue system[3] implementing ISU with only one element of the IS – `$lm` (latest move). `Interpret` state implements speech recognition and can raise `SAYS` event with a recognised value; `Generate` state implements speech synthesis which speaks out the value of `NEXT_MOVE` event. The `Process` state implements dialogue management and is initiated with `$lm` containing an initial move "ping"; it starts in a `Select` state and, if `$lm` is not empty, raises a `NEXT_MOVE` event with a value of `$lm` and transitions to `Update` state. Then, upon receiving `SAYS` event it updates `$lm` with the event value and transitions back to `Select` state.

SISU is strongly inspired by Kronlid and Lager (2007) system, which used a version of State Chart

[2] https://studio.talkamatic.se

[3] Previously used by Bos et al. (2003); Kronlid and Lager (2007) to illustrate their implementations of ISU.

XML (SCXML) extended with Prolog-style conditions for state transitions. Our implementation uses XState[4], a state management library for JavaScript and TypeScript which implements Harel's formalism and uses actor model for concurrency. Despite an abundance of commercially available dialogue-building software based on FSMs and deterministic flows, such as Dialogflow[5], statecharts are supported by the World Wide Web Consortium (W3C) specification[6], which serves as a guidance for modern implementations, such as XState.

3 Architecture

In this section, we describe the general architecture of SISU, exemplified by a simple form-filling dialogue application in the context of scheduling.

3.1 Information state

For representing the information state, we utilise the data storage of XState, and declare the IS as a type:

```
type InformationState = {
  next_moves: Move[];
  domain: Domain;
  database: Database;
  private: {
    agenda: Action[];
    plan: Action[];
    bel: Proposition[] };
  shared: {
    lu: { speaker: Speaker; moves: Move[] };
    qud: Question[];
    com: Proposition[]; };};
```

The information state is composed of the following data structures:

moves Moves are essentially dialogue acts that can be performed by the user or the system. In SISU, they consist of a type (e.g., "answer" or "ask") and content.[7] For example, the type of the "ask" move is defined as follows:

```
type AskMove = { type: "ask";
                 content: Question; };
```

questions There are several types of questions that can be supported by the system, i.e. wh-questions which contain a predicate to be fulfilled.

```
type WhQuestion = { type: "whq";
                    predicate: string };
```

For instance, a question concerning the location of a booking can be represented as `{type: "whq", predicate: "booking_room"}`.

propositions A predicate combined with an argument forms a proposition. For instance, a proposition that a lecture takes place in G212 can be represented as `{predicate: "booking_room", argument: "G212"}`.

plans and actions Plans are domain-specific and high-level descriptions of how goals are achieved. Plans are represented as lists of actions. For example, a plan for responding to a question concerning the location of a booking can be represented as:

```
{ type: "issue",
  content: { type: "whq",
             predicate: "booking_room" },
  plan: [{ type: "findout",
           content: {
             type: "whq",
             predicate: "booking_course" }},
         { type: "findout",
           content: {
             type: "whq",
             predicate: "booking_day" }},
         { type: "consultDB",
           content: {
             type: "whq",
             predicate: "booking_room"}}]}
```

This plan enables the system to respond to questions concerning locations of a booking by first finding out which course and day the question concerns, and by then consulting a database.

3.2 Update rules

In SISU, update rules have the type

```
type Rule =
   (context: InformationState) =>
     ((x: void) => InformationState) | undefined;
```

which takes an information state as input and, depending on the truth value of the precondition, either, if the precondition is not met, returns `undefined`, or else a function that returns the updated state.[8] For instance, the following rule, which implements rule 2.2 from Larsson (2002), pushes a question recently uttered by the system, if such one exists, onto the stack of questions under discussion (QUDs).

[4] https://stately.ai/docs/xstate

[5] see Sabharwal and Agrawal (2020) and https://cloud.google.com/dialogflow/cx/docs/basics

[6] https://www.w3.org/TR/scxml/

[7] There is some similarity with the notion of intents, but those are typically domain-specific, whereas in our framework we assume that the update rules can be domain-general which requires operating over more abstract data structures.

[8] XState's `assign()` method is used when the rule is instantiated.

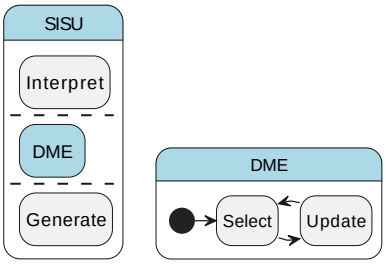


Figure 2: SISU and DME statecharts (zoomed-out).

```
integrate_sys_ask: ({ is }) => {
  if (is.shared.lu!.speaker === "sys") {
    for (const move of is.shared.lu!.moves) {
      if (move.type === "ask") {
        const q = move.content;
        return () => ({
          ...is, shared: {
            ...is.shared,
            qud: [q, ...is.shared.qud]
          }});}}};
```

We make use of spread (...) JavaScript syntax, which effectively allows overriding parts of information state as well as operating stacks (i.e. pushing a question Q into the stack of QUDs).[9]

3.3 Dialogue Move Engine (DME)

Overall architecture of SISU (statechart on Figure 2) extends the "parrot" example with a Dialogue Move Engine (DME) statechart. It is responsible for updating the IS and selecting moves to be produced. The structure of DME reproduces the statechart introduced by Kronlid and Lager (2007).

Figures 3 and 4 show move selection and state update processes. We use syntactic sugar "(/*condition*)" to represent an update rule which consists of an update *action* performed in case when the *condition* is met. The update action and evaluator for the condition are provided as functions of the type introduced in Section 3.2. If no update rules can be applied in a given state, it transitions to the next one (depicted with arrows without conditions or event), except for the `Grounding` state in which DME waits for the `SAYS` event (containing either a recognised move or a move produced by the system) which triggers further update process. When the move selection is done, the move is taken up by the language generation state orthogonal to DME (similarly to Figure 1).

[9] Ginzburg (2012); Cooper (2022) use asymmetric merge operation for expressing updates of an information state. For JavaScript object literals, when one object is spread into another, the property can be overridden by the last assigned value.

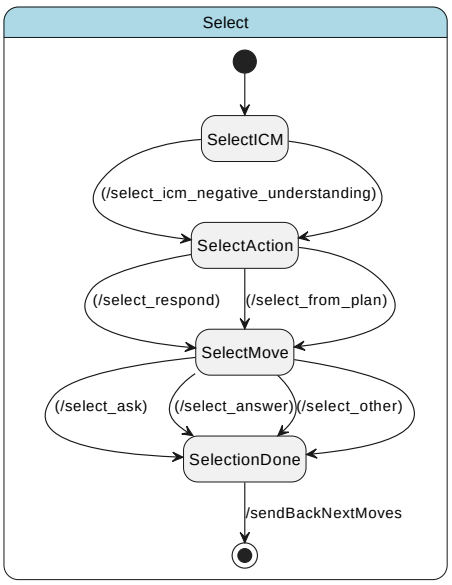


Figure 3: Zoomed-in `Select` statechart.

3.4 Domain knowledge and database

Following Larsson and Berman (2016), SISU separates general knowledge about dialogue from domain-specific knowledge. The developer (or a LLM, as we will show in Section 4) supplies the latter, e.g. dialogue plans and API integrations.

4 Pre-generating dialogue domains with LLMs

Despite their sometimes impressive performance, LLMs are associated with a host of well-known problems (hallucinations, bias etc.) deriving from the overall problem of controlling the behaviour of LLMs (Kann et al., 2022). The absolute majority of methods for dealing with this problem is of the "guardrails" type. In LLM-based Conversational AI, however, the user is still interaction with an LLM at runtime, and it is difficult or impossible to guarantee that guardrails always work. Ayyamperumal and Ge (2024) discuss various guardrail approaches such as layered protection models, system prompts, Retrieval-Augmented Generation (RAG) architectures and bias mitigation, and observe that "[c]rucial challenges remain in implementing these guardrails." Xu et al. (2024) show that hallucination is not just a temporary glitch, but are in fact inevitable in LLMs.

An alternative to using LLMs is of course to manually build dialogues, as done in e.g. Dialogflow (Sabharwal and Agrawal, 2020) and many other toolkits. In the SISU approach, we call such dialogue specifications *dialogue domain descriptions*,

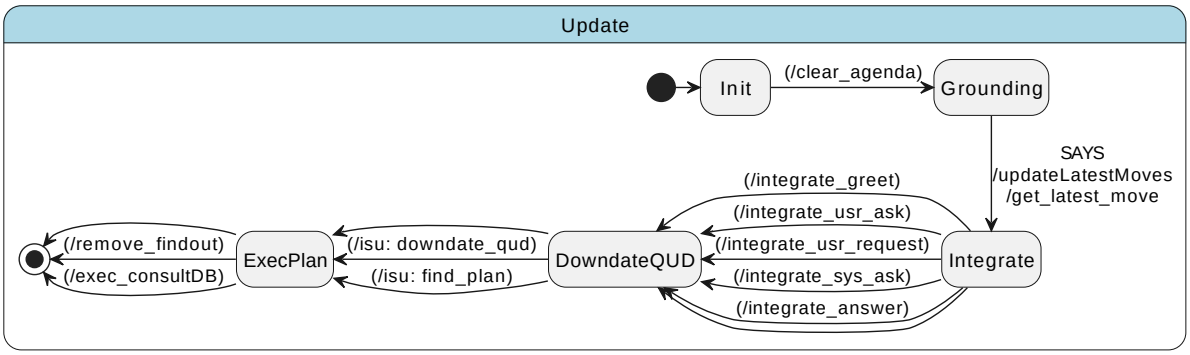


Figure 4: Zoomed-in `Update` statechart.

or dialogue domains for short. Although dialogue domains are concise and expressed on a high level of abstraction, coding them by hand can be time- and resource-intensive. Furthermore, the formalism can be challenging for non-technical domain experts. Pre-Generative Conversational AI (Larsson, 2024) uses LLMs for pre-generating domain knowledge to address these problems. Specifically, the LLM is fed relevant information such as input type definitions and example dialogues between system and user, and returns a dialogue domain and database integration. The interaction between an LLM and dialogue developer can be included in a workflow alongside user testing, enabling insights from testing to be fed to the LLM to inform subsequent improvements. We provide the details of our approach in Appendix A, namely, how an LLM can be used to generate domain and database for a small scheduling scenario.[10]

5 Coding low-level dialogue routines with statecharts

In addition to FSMs being a prominent framework for building dialogue systems (McTear, 2020), previous studies explained complex natural language grounding phenomena using FSMs, such as vocabulary enquiries in language tutoring (del Fresno et al., 2022) and spelling out names (Howes et al., 2019; Larsson et al., 2020). Such FSMs include routinised adjacencies between dialogue acts in human dialogue; to build a dialogue system based on these definitions one must include considerations from system perspective and cast FSMs into a statechart.[11] Figure 8 in Appendix B shows an example a statechart based on a FSM from Larsson et al. (2020). SISU can include routinised procedures defined in such way and involve ISU only when certain information is grounded, e.g., when the user and the system have demonstrated agreement on a particular spelling of a word.

6 Conclusions

In this paper we described a framework combining statecharts and ISU approach. We described its architecture and illustrated it with a small scheduling scenario. We underlined two main advantages of our approach, first, its adequacy for pre-generating dialogues with LLMs and, secondly, the possibility of mixing routinised dialogue procedures, such as task-specific grounding described with FSMs or statecharts, with ISU based on theoretical principles.

Future work will include empirical evaluation of the framework as well as expanding the case study to cover a range of more complex domains.

Acknowledgements

Vladislav Maraev was supported by Swedish Research Council (VR) grant 2023-00358 – Social laughter for virtual agents (SOCLAVA). We also acknowledge support from the Swedish Research Council VR project 2014-39 for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

[10] An industry strength implementation of Pre-Generative Conversational AI is available in Talkamatic Studio. The examples in this paper are illustrative toy examples intended to convey the gist of the idea, and do not reflect how dialogue pre-generation is implemented in Talkamatic Studio.

[11] Statecharts can be also used for describing routines in human dialogue, because they allow less clutter thanks to state hierarchy.

References

Suriya Ganesh Ayyamperumal and Limin Ge. 2024. Current state of LLM risks and AI guardrails. *arXiv preprint arXiv:2406.12934*.

Johan Bos, Ewan Klein, Oliver Lemon, and Tetsushi Oka. 2003. DIPPER: Description and formalisation of an information-state update dialogue system architecture. In *Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue*, pages 115–124.

Jenny Brusk. 2008. Dialogue management for social game characters using statecharts. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ACE '08, page 219–222, New York, NY, USA. Association for Computing Machinery.

Ronnie Cann, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

Robin Cooper. 2022. *From Perception to Communication: a Theory of Types for Action and Meaning*. Oxford University Press, Oxford. In press.

Andrea Carrión del Fresno, Staffan Larsson, and Vladislav Maraev. 2022. Dialogue strategies for... cómo se dice entrenamiento de vocabulario? In *Proceedings of the 26th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Dublin, Ireland. SEMDIAL. 10 pages.

Jonathan Ginzburg. 2012. *The interactive stance*. Oxford University Press, Oxford.

David Harel. 1987. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274.

Christine Howes, Anastasia Bondarenko, and Staffan Larsson. 2019. Good call! Grounding in a Directory Enquiries Corpus. In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue*, London, United Kingdom. SEMDIAL.

Katharina Kann, Abteen Ebrahimi, Joewie Koh, Shiran Dudy, and Alessandro Roncone. 2022. Open-domain dialogue generation: What we can do, cannot do, and should do next. In *Proceedings of the 4th Workshop on NLP for Conversational AI*. Association for Computational Linguistics.

Fredrik Kronlid and Torbjörn Lager. 2007. Implementing the information-state update approach to dialogue management in a slightly extended SCXML. In *Proceedings of the 11th International Workshop on the Semantics and Pragmatics of Dialogue (DECALOG)*, pages 99–106.

Staffan Larsson. 2002. *Issue-based dialogue management*. Department of Linguistics, Göteborg University.

Staffan Larsson. 2024. Pre-generative conversational AI. In *Proceedings of the 28th Workshop on the Semantics and Pragmatics of Dialogue - Poster Abstracts*, Trento, Italy. SEMDIAL.

Staffan Larsson and Alexander Berman. 2016. Domain-specific and general syntax and semantics in the talkamatic dialogue manager. *Empirical Issues in Syntax and Semantics*, 11:91–110.

Staffan Larsson, Christine Howes, and Anastasia Bondarenko. 2020. Could you spell that again please? Towards a formal model of grounding in directory enquiries. In *First AISB Symposium on Conversational AI (SoCAI)*.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *NLE Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering*, pages 323–340.

Michael McTear. 2020. *Conversational AI: Dialogue systems, conversational agents, and chatbots*, volume 13. Morgan & Claypool Publishers.

Gregor Mehlmann, Birgit Endraß, and Elisabeth André. 2011. Modeling parallel state charts for multithreaded multimodal dialogues. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, page 385–392, New York, NY, USA. Association for Computing Machinery.

Navin Sabharwal and Amit Agrawal. 2020. Introduction to Google Dialogflow. In *Cognitive virtual assistants using google dialogflow: develop complex cognitive bots using the Google Dialogflow platform*, pages 13–54. Springer.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: A review. *Computer Speech & Language*, 67:101–178.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

A Pre-generating a dialogue domain with ChatGPT

In Pre-generative Conversational AI (Larsson, 2024) implemented on industrial level in Talkamatic Studio, dialogue domain data is generated by an LLM from content such as a database API or a text. The TDM uses this data to enable flexible dialogue in various types of dialogue, including educational, customer service, instructional and negotiative dialogue.

Here, as a simple demonstration of the feasibility of pre-generating dialogue domains using LLMs, we feed a dialogue example to ChatGPT (`GPT-4o`), together with the type definition for a domain and database, and ask it to generate a domain and database that supports the dialogue (see Figure

5). The model's generated domain and database are included in Figures 6 and 7 respectively[12].

Automated testing of the system's resulting behaviour validates that the model's generated code works as intended, without any errors. It is worth noting that except for a minimal amount of comments in the type definition, no documentation of the dialogue domain formalism or the overall dialogue system is provided.

One can also note that the code generated by an LLM contains a semantic peculiarity: all predicates and individuals are declared as a single sort (`course`). This can seem unintuitive, since days and locations are not courses. One consequence of this peculiarity is that the answer "Thursday" will be considered relevant in relation to the question "Which course?". This problem can potentially be addressed within an LLM-based development framework by extending the set of dialogue examples fed to the LLM.

B Coding parts of dialogue with statecharts

Figure 8 shows an example of how procedural FSM for name-spelling derived from human-human dialogue (Howes et al., 2019; Larsson et al., 2020) can be adapted to statechart definition for further use in SISU-based dialogue system.

[12]Note that this is an illustrative toy example and does not reflect how dialogue pre-generation is implemented in Talkamatic Studio.

```
Create a domain and database that supports the dialogue below between user
(U) and system(S), and that adheres to the type definition below.

SUPPORTED DIALOGUE

S: Hello! You can ask me anything!
U: Where is the lecture?
S: Which course?
U: Dialogue Systems 2
S: Which day?
U: Thursday
S: The lecture is in G212.

TYPE DEFINITION

type Individuals = Predicates;
type Predicates = { [index: string]: string };
export type Domain = {
  plans: PlanInfo[];
  predicates: Predicates;  // Mapping from predicate to sort
  individuals: Individuals; // Mapping from individual to sort
};

export type PlanInfo = {
  type: "action" | "issue";
  content: null | Proposition | ShortAnswer | Question;
  plan: Action[];
};

export type Database = {
  consultDB: (q: Question, p: Proposition[]) => Proposition | null;
};

export type ShortAnswer = string;
export type Proposition = {
  predicate: string;
  argument: string;
};

export type Question = WhQuestion;
type WhQuestion = { type: "whq"; predicate: string };

export type Action = {
  type:
    | "greet"
    | "respond" // not to be used in plans
    | "raise"
    | "findout"
    | "consultDB";
  content: null | Question;
}
```

Figure 5: Input fed to ChatGPT.

```
export const domain: Domain = {
  predicates: {
    course_day: "course",
    course_location: "course",
    course_name: "course",
  },
  individuals: {
    "Dialogue Systems 2": "course",
    "Thursday": "course",
    "G212": "course",
  },
  plans: [
    {
      type: "issue",
      content: { type: "whq", predicate: "course_location" },
      plan: [
        { type: "findout", content: { type: "whq", predicate: "course_name" } },
        { type: "findout", content: { type: "whq", predicate: "course_day" } },
        { type: "consultDB", content: { type: "whq", predicate: "course_location" } }
      ]
    }
  ]
};
```

Figure 6: Domain generated by ChatGPT.

```
export const database: Database = {
  consultDB: (q, props) => {
    if (
      q.type === "whq" &&
      q.predicate === "course_location" &&
      props.some(p => p.predicate === "course_name" && p.argument === "Dialogue Systems 2") &&
      props.some(p => p.predicate === "course_day" && p.argument === "Thursday")
    ) {
      return { predicate: "course_location", argument: "G212" };
    }
    return null;
  }
};
```

Figure 7: Database generated by ChatGPT.

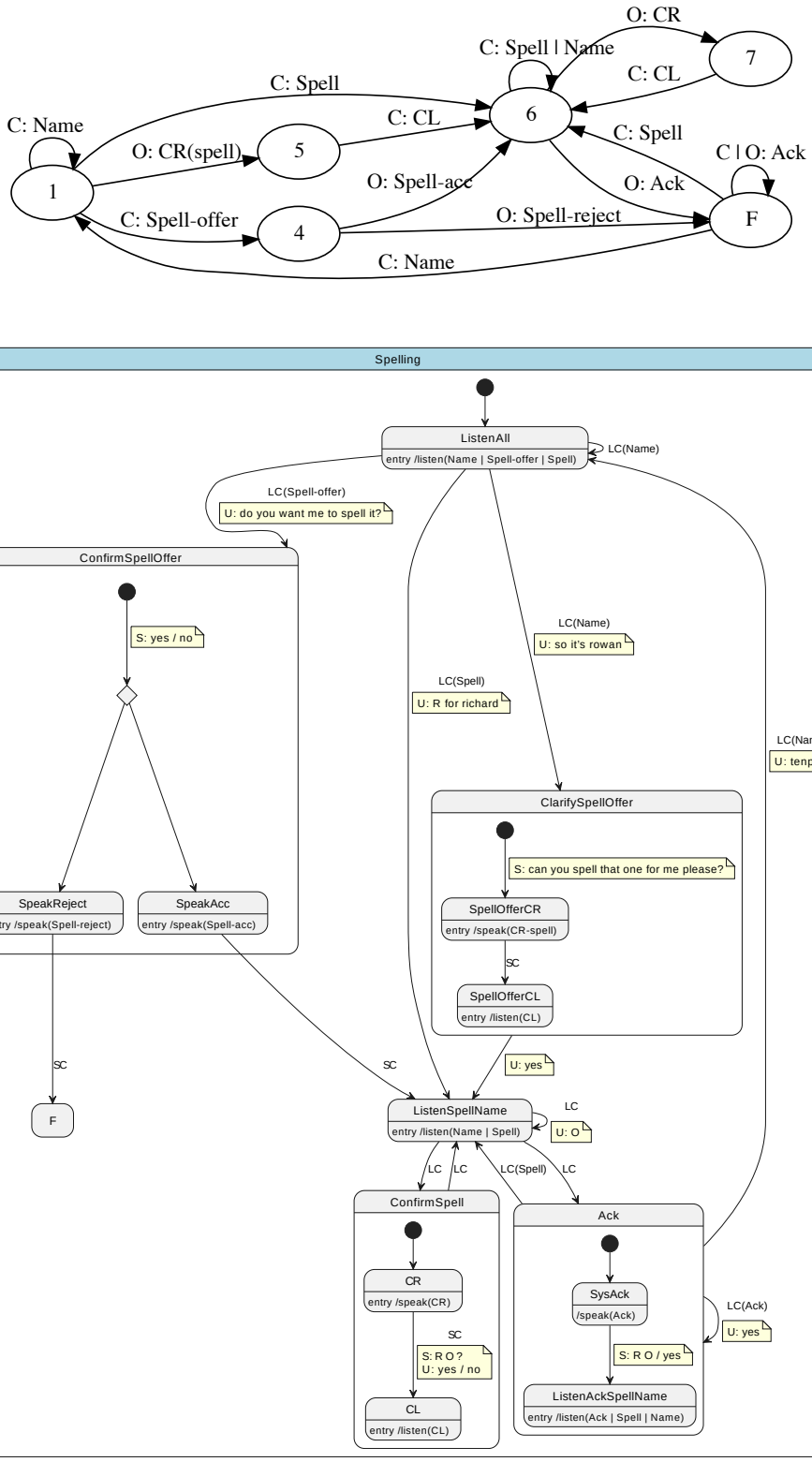


Figure 8: **Above**: name-spelling FSM adapted from Larsson et al. (2020). **Below**: a corresponding statechart. SC = system yields the turn; LC = user yields the turn. Yellow notes exemplify possible utterances.

Towards Neuro-Symbolic Approaches for Referring Expression Generation

Manar Ali[*1,2], Marika Sarzotti[*2,4], Simeon Junker[1,3],
Hendrik Buschmeier[1,2], Sina Zarrieß[1,3]
[1]CRC 1646 'Linguistic Creativity in Communication', Bielefeld University, Germany
[2]Digital Linguistics Lab, Bielefeld University, Germany
[3]Computational Linguistics Group, Bielefeld University, Germany
[4]Center for Mind/Brain Sciences (CIMeC), University of Trento, Italy
{manar.ali|simeon.junker|hbuschme|sina.zarriess}@uni-bielefeld.de
marika.sarzotti@studenti.unitn.it

Abstract

Referring Expression Generation (REG) has a long-standing tradition in computational linguistics, and often aims to develop cognitively plausible models of language generation and dialogue modeling, in a multimodal context. Traditional approaches to reference have been mostly symbolic, recent ones have been mostly neural. Inspired by the recent interest in neuro-symbolic approaches in both language and vision, we revisit REG from these perspectives. We review relevant neuro-symbolic approaches to language generation on the one hand and vision on the other hand, exploring possible future directions for cognitively plausible models of reference generation/reference game modeling.

1 Introduction

Referring Expression Generation (REG) in visual scenarios is a traditional and widely studied task in cognitively motivated work on Natural Language Generation (NLG). At its core, the task consists of generating an expression that refers to a visual object in a given scene, in a way that an addressee can identify the intended target (Reiter and Dale, 2000). Although this task may seem basic and constrained at first, it is multifaceted and involves overcoming several implicit or explicit challenges at the intersection of language and vision. These challenges include segmenting and understanding the low-level visual input (*visual processing*), determining the properties of the referential target that distinguish it from all distractors (*content determination*), and, finally, formulating the conceptual information into well-formed linguistic expressions (*linguistic realization*), see Schüz et al. (2023).

Existing research in REG has approached this problem using two different methodologies, see Figure 1: The landscape can be roughly divided into *symbolic* and *neural* (or *visual*) approaches,

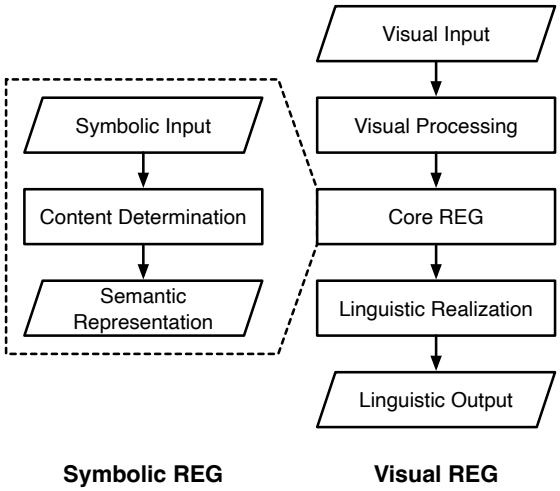


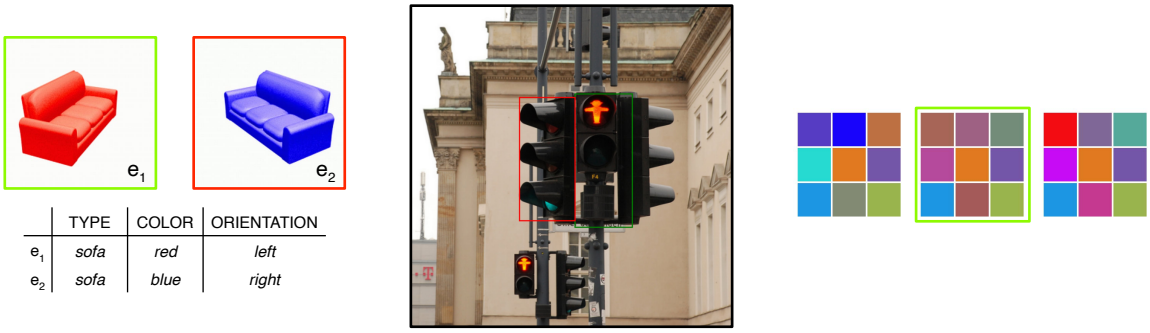
Figure 1: Conceptual illustration of processing steps in common models for symbolic and neural-visual Referring Expression Generation (REG; Schüz et al., 2023). While symbolic REG focuses primarily on selecting discriminative properties of the target, low-level inputs and natural language outputs require further processing steps or more general methods.

each with their own characteristics. Symbolic methods offer controllable, transparent, and cognitively plausible ways of pragmatic reasoning, but most approaches focus on specific challenges (i.e., content determination), and it is difficult to apply the algorithms to natural scenarios due to their dependence on symbolic inputs. In contrast, neural methods can be easily applied to more natural or complex scenarios, as the systems are trained end-to-end, implicitly learning all the necessary steps from visual processing to linguistic realization. However, neural approaches are notoriously difficult to control, their cognitive plausibility is debatable, and the exact processing methods are generally concealed due to the black-box nature of neural systems.

Against this background, neuro-symbolic approaches in computational linguistics and NLP are currently attracting considerable research inter-

[*]These authors share first authorship.

Proceedings of the 2025 CLASP Conference on Language models And RePresentations (LARP), pages 38–50



(a) Symbolic input (b) Natural visual input (c) Abstract visual input

Figure 2: Examples for different REG settings. Settings like (a) (van Deemter et al., 2006) have been traditionally addressed with symbolic approaches, whereas the settings in (b) (Kazemzadeh et al., 2014) and (c) (McDowell and Goodman, 2019) call for (partially) neural approaches, due to the lack of symbolic input representations.

est: By combining neural and symbolic processing methods, it becomes possible to build systems that retain the flexibility and performance of neural systems, but become more robust, controllable, and transparent (Hamilton et al., 2024).

In this paper, we review existing symbolic and neural approaches to REG and discuss how these lines of work can be integrated using neuro-symbolic approaches. We argue that symbolic processing methods can be applied to different stages in a neural REG processing pipeline, potentially leading to more transparent, cognitively plausible, and robust REG systems. What exactly is considered a neuro-symbolic system is not always consistently defined. Here, we treat approaches as neuro-symbolic that include neural modeling components, but also methods for reasoning about symbolic information.

2 Background

2.1 Referring Expression Generation

Symbolic REG Generating references to objects has been a long-standing field of interest in computational linguistics see Krahmer and van Deemter 2012 for a survey). Influential work (e.g., Dale 1989; Dale and Haddock 1991) started to focus on algorithms for *content selection*, comparing properties of a target object with potential distractors to determine a set of properties that can be formulated into discriminative descriptions. Building on a Gricean notion of pragmatics (Grice, 1975), algorithms are considered successful if they provide sufficient information to identify the intended referent without being overly informative. A prime example is the Incremental Algorithm (Dale and Reiter, 1995), which iterates through attributes in a defined order of preference, selecting those that rule out any distractors until only the target remains.

Subsequent work extended the scope by including, e.g., relational descriptions (Krahmer and Theune, 2002; Krahmer et al., 2003), references to sets of multiple targets (Horacek, 2004; Gatt and van Deemter, 2007), or notions of prominence or salience to pre-select contextually relevant distractors (Kelleher and Kruijff, 2006; Belz et al., 2010).

Much of the work in symbolic REG consists of deterministic, rule-based search algorithms for content determination that operate on symbolic knowledge bases (see Figure 2a). However, there are alternative approaches such as the probabilistic PRO (van Deemter et al., 2012) and RSA (Frank and Goodman, 2012) models or the graph-based algorithm in Krahmer et al. (2003). Further approaches combine content determination with linguistic realization (Horacek, 1997; Stone and Webber, 1998; Siddharthan and Copestake, 2004), see van Deemter (2016). However, the reliance on symbolic input information remains as a characteristic feature.

Neural REG Visual environments are commonly used as prime examples or application domains for symbolic REG algorithms, but the reliance on symbolic inputs largely prohibits direct application in natural visual scenarios where this information is not available (Schüz et al., 2023). In recent years, work on *visual REG* has reformulated the task as an *image-to-text* generation problem, enabled by corpora such as RefCOCO (Kazemzadeh et al. 2014; see Figure 2b) and more general advances in neural vision-and-language modeling. Here, the goal is to generate descriptions from raw visual representations of objects in natural images.

Similar to image captioning (Vinyals et al., 2015),

neural REG models are commonly trained end-to-end and follow the encoder–decoder scheme, where raw visual inputs are transformed into intermediate representations by an image encoder and then passed to a language decoder. Hence, neural approaches to visual REG differ fundamentally from their symbolic counterparts: Low-level perceptual inputs replace the high-level symbolic information, and while symbolic approaches often focus on content determination, neural systems cover all steps from visual processing to linguistic realization, although the exact processes are largely concealed in the connectionist structures of the neural systems.

Much of the existing work revolves around methods to optimize the discriminative power of generated expressions (see Schüz et al., 2023), for example by including different simulations of addressee behaviour (Mao et al., 2016; Luo and Shakhnarovich, 2017; Yu et al., 2017; Schüz and Zarrieß, 2021), enriching visual input representations with discriminative information (Yu et al., 2016; Liu et al., 2017) or directing systems to pragmatically relevant features (Li and Jiang, 2018; Tanaka et al., 2019; Liu et al., 2020; Kim et al., 2020; Sun et al., 2022). Other works focus on aspects beyond discriminativeness, e.g., iteratively refined expressions (Zarrieß and Schlangen, 2016; Ye et al., 2023), effects of decoding methods (Zarrieß and Schlangen, 2018), generating diverse expressions (Panagiaris et al., 2020, 2021), REG in visual dialogue (Willemsen and Skantze, 2024) or the role of visual scene context (Junker and Zarrieß, 2024). More recently, work on neural REG has started to incorporate vision-language models (Bracha et al., 2023; Guo et al., 2024; Liang et al., 2024) and referring expressions have been included in multitask frameworks (Wang et al., 2022b; Lu et al., 2023; You et al., 2023; Xiao et al., 2024), although with focus on the inverse referring expression comprehension task.

2.2 Neuro-symbolic approaches in general

Neuro-Symbolic AI is a growing research field concerned with the development of AI systems which should be able to simulate and integrate the two cognitive processes commonly considered as the core of intelligent behaviour, namely the ability to learn from experience and to reason on what has been learned (Valiant, 2003). Researchers have been trying to pursue this goal by combining neural networks and deep learning methods, excellent at handling parallel computation, unstructured data, and pattern recognition, with purely symbolic approaches, typically leveraging formal logic or structured representations, which are verifiable and data-efficient, and allow structured and logical reasoning about data and its patterns (Garcez and Lamb, 2023; Hamilton et al., 2024).

The problem of how neural networks can handle and represent symbolic knowledge has been present in the literature since early attempts to computationally model brain processes (Bader and Hitzler, 2005). Over the past decade, however, it was Deep Learning that got the most attention in research and application. Lately, it was argued that in order to achieve rich, semantically sound and explainable AI systems, research efforts should focus on the integration between methods affording reasoning abilities and Deep Learning (Garcez and Lamb, 2023), resulting in a new interest in neuro-symbolic integration. To clarify and systematize the work on neuro-symbolic integration, highlighting similarities and differences among the various contributions, taxonomies have been devised. The most well-known was proposed by Kautz (2022) and further streamlined in Hamilton et al. (2024):

Sequential Sequential architectures are current the dominant approach in Deep Learning when the input and output of neural networks are symbolic in nature, such as in the case of Natural Language Processing, where symbolic inputs, namely words and word sequences, are converted into vectors and processed by a neural network.

Nested Nested architectures are those that loosely couple a symbolic reasoning system, such as a problem solver or a planner, with a neural component that will guide certain decision processes. One instance is DeepMind's AlphaGo (Silver et al., 2016), where a Monte Carlo tree search algorithm is paired with a neural network tasked to evaluate game states and suggest moves.

Cooperative Cooperative architectures include a neural component which receives raw inputs, such as images' pixels, and converts them into symbolic data structures, for instance graphs or logic-based representations, which will be used by a symbolic reasoner. One example system is DeepProbLog (Manhaeve et al., 2018), which involves a neural network which parametrizes the truth distribution of predicates with respect to an input, and a probabilistic logic program for reasoning with them.

Compiled Compiled architectures are tightly coupled approaches, as there is no modular sub-division to handle learning and reasoning. In fact, these systems involve standard neural networks undergoing training regimes based on symbolic rules, by having knowledge compiled into the training set or the network's weights, or enforced via specific optimization functions. They are instantiated by Logic Neural Networks (Riegel et al., 2020), where symbolic rules are embedded directly into the architecture, as neurons in the network's layers represent specific logical operations, and Logic Tensor Networks (Badreddine et al., 2022), which are optimized to maximize the satisfiability of grounded (represented as real-valued tensors) formulas.

Ultimately, a neuro-symbolic system could have a fully integrated architecture where the symbolic reasoning component is embedded in the neural one. Hamilton et al. (2024) include this potential architecture in the *Nested* class, though, to this day, there are no implemented solutions that truly embody this definition.

3 Neuro-symbolic approaches to REG

Encoder-decoder models in vision-language generation tasks like REG always combine neural and symbolic aspects, as they map raw inputs (images) to symbolic outputs (text). However, in most approaches for visual REG (Section 2.1) the transformation from perceptual to symbolic information takes place at the very end of the processing pipeline and merely consists of a final mapping over the model's vocabulary during inference, without any reasoning processes involving those symbolic units. In this section we describe existing approaches for reference generation that go beyond this level of neuro-symbolic integration, and include further sources of symbolic information or symbolic reasoning processes.

Chamorro-Martínez et al. (2021) propose a system for referring expression generation (REG) that combines deep learning with symbolic processing. They use a Mask R-CNN model to segment images and detect objects with associated confidence scores. Fuzzy modeling is then applied to derive color attributes and spatial relationships between objects, which are represented in a graph structure—nodes represent objects with category and color labels, and edges represent spatial relations, all annotated with fuzzy confidence values. This symbolic graph is used by a content selection algorithm to identify the most discriminative properties for referring to each object.

Tsvilodub et al. (2024) present a neuro-symbolic Iterative Model (IM) for referring expression generation, inspired by the Incremental Algorithm (Dale and Reiter, 1995). The model combines large language models (LLMs) with symbolic reasoning. An LLM-based utterance proposer generates simple candidate descriptions, which a second LLM module evaluates for semantic adequacy. A symbolic contrastivity selector then assesses how well each description distinguishes the target from distractors. If no maximally contrastive expression is found, the process iterates by adding more details. Designed for visual tasks, the model avoids processing raw visual input by working with verbal scene descriptions.

In Junker and Zarrieß (2024), the low-level target representations used as input in their encoder-decoder models are supplemented by symbolic *scene summaries* that represent the relative area in the visual context covered by different types of objects, in order to support the robustness of referring expressions under visually challenging conditions. The results show that by including scene-level symbolic information, the models can correctly infer the type of the target object, even when visual representations of the target are severely distorted.

Apart from those works, the Rational Speech Acts framework (RSA; Frank and Goodman, 2012; Frank et al., 2016) emerges as the most prominent approach for integrating neural processing and symbolic reasoning. Here, generally, Bayesian inference is used to model pragmatic behaviour, in terms of rational speakers (S_1) that reason about how literal listeners (L_0) would understand utterances produced by literal speakers (S_0).

Andreas and Klein (2016) propose an approach for generating contrastive scene descriptions in a reference game involving visual scenes as targets and distractors. Ignoring distractor context, a neural language model acting as the literal speaker S_0 takes encoded images and produces descriptions of them. A neural literal listener L_0 takes an image description and a set of possible referents and produces a distribution over candidate scenes, for each indicating the probability that this scene is the referent described. Finally, a RSA reasoning speaker S_1 ties those models together by drawing a set of samples from S_0 and using Bayesian inference to select a description scored high by both S_0 and L_0. Similar to Tsvilodub et al. (2024), this

system relies on symbolic feature representations for objects depicted in the scenes.

In their work on pragmatically informative image captioning, Cohn-Gordon et al. (2018) follow the same intuition, but apply the pragmatic reasoning at each step of the iterative inference process. Here, S_0 is a character-level image captioning model, consisting of a CNN encoder and an LSTM decoder. At each decoding step, S_0 outputs a probability distribution over possible continuations of a partial caption consisting of the start token in the initial run. For each possible continuation, L_0 returns a distribution over potential target images. Finally, S_1 takes the L_0 distribution over images and re-weights the S_0 predictions for possible continuations by L_0's ability to infer the correct target image with this continuation.

The decoding algorithm in Vedantam et al. (2017) pursues the same idea, but with word-level captioning models and without the recursive back-and-forth between the speaker and listener agents as defined in the RSA model.

Several papers in REG have adopted the idea of performing pragmatic reasoning during the inference of otherwise context-agnostic generation models: Schüz and Zarrieß (2021) directly apply this approach to REG using the discriminative decoding methods from Cohn-Gordon et al. (2018) and Vedantam et al. (2017), but define targets and distractors as objects within a single image rather than as separate images. Here, at first, the bounding box content for a visual target object is encoded and passed to the model decoder. During decoding, output probabilities are compared at each step with the predictions of the same model when processing distractor objects instead of the target. On this basis, the token probabilities for the target are adjusted in favor of words that have a higher probability for the target than for distractors. In line with findings from image captioning (Schüz et al., 2021), the authors show that this method increases both the pragmatic informativeness and the linguistic diversity of generated expressions.

Zarrieß and Schlangen (2019) use a similar method to reason about possible categorizations of target objects, assuming that very specific terms should be avoided when models are uncertain about object categories. Again, they incorporate RSA-style reasoning into the iterative decoding process. However, their model does not reason about which words are informative for identifying the target, but about which terms should be used for the target to avoid erroneous descriptions, given the uncertainty about object categories. They show that their model generates more expressions without any nouns or category labels, consistent with the hypothesized strategies for describing unknown objects. With respect to an external listener model, the proposed strategy increases the resolution accuracy for most categories of objects.

Finally, White et al. (2020) consider further possibilities for how the Rational Speech Acts framework can be incorporated into neural generation models. In addition to a *full RSA* model, which includes an exhaustive reasoning process, where all possible utterances are tested for how effectively they allow the trained listener model to identify the target, they also consider a *sample re-rank* model which resembles Andreas and Klein (2016)'s approach in that a smaller number of candidate utterances are sampled from the speaker model and then re-ranked by the listener. In addition, they present a model that *amortizes* the computational costs of exhaustive RSA reasoning by directly optimizing a speaker model with respect to the utterances that an RSA model would prefer. To this end, during training, at each optimization step an utterance is sampled from the speaker model to be trained, which is then evaluated by the listener model and translated into training signals depending on its communicative success. This transfers the symbolic reasoning process from the inference to the training stage; the subsequent decoding process can thus be carried out using computationally more efficient methods. The results show that the amortized model almost achieves the pragmatic effectiveness of the full RSA model, but is significantly more efficient.

Overall, neuro-symbolic processing remains an exception in REG and related tasks. Apart from Junker and Zarrieß (2024), symbolic components generally target the linguistic level rather than the visual processing of inputs. Most commonly, RSA or related approaches are used to reason about the pragmatic informativeness of linguistic symbols (characters, words, or sentences), sometimes as part of the training procedure (White et al., 2020). Similar to content selection in symbolic REG, Chamorro-Martínez et al. (2021) and Tsvilodub et al. (2024) employ similar procedures at a more conceptual level, i.e., with regard to the question of which attributes best describe the referent, regardless of the concrete realization.

Regarding Hamilton et al. (2024)'s taxonomy, the approaches can be placed at different levels:

The addition of symbolic inputs renders Junker and Zarrieß (2024) a *sequential* system, while Tsvilodub et al. (2024) can be seen as *nested* with symbolic components controlling the entire process. Chamorro-Martínez et al. (2021) and inference-level RSA variants are *cooperative* because deep learning methods form the basis for symbolic reasoning. Finally, White et al. (2020)'s amortized model is a *compiled* system where symbolic reasoning is integrated into the training regime.

4 Neuro-symbolic NLG and Vision

Only a few approaches in REG surpass a level of neuro-symbolic integration that is trivial for vision-language generation tasks. This section will therefore discuss some neuro-symbolic approaches in two REG-relevant fields: NLG more generally and visual processing in vision-language tasks.

4.1 Natural Language Generation

Graph-based methods One approach is to integrate structured data representations, such as knowledge graphs, into the language generation process. This is often referred to as knowledge injection, where knowledge from external sources is incorporated into models to improve their output quality (Cadeddu et al., 2024). Knowledge graphs represent general-purpose, or domain-specific (Ji et al., 2022) data as nodes (entities) and edges (relations), a flexible and powerful way of encoding knowledge.

Knowledge graphs have been used in various NLG tasks (see Panchendrarajan and Zubiaga 2024 for a survey). In language modeling, knowledge graphs can be used by converting them into vector representations using graph embedding methods and feeding them as input to a language model. Other models adapt existing text-generation models to generate text directly from knowledge graphs (e.g., Koncel-Kedziorski et al., 2019). Knowledge graphs have also been used in dialogue systems. For instance, Zhang et al. (2020) proposed a method that constructs concept graphs from dialogue inputs and expands them to include related one-hop and two-hop concepts from a commonsense knowledge base. These graphs are then encoded into vector representations using a graph neural network. The resulting vectors are integrated with the original input to incorporate external knowledge and guide the model in generating coherent responses. Likewise, knowledge graphs have been used in text summarization tasks, where faithfulness to the original text is essential. Some models (e.g., Wang et al., 2022a) introduce a knowledge graph pipeline that extracts relational triplets from the source text and encodes them using graph embeddings. A filtering step uses a trained classifier to identify key facts from the source by predicting their importance. This allows the model to focus on salient and relevant information. The filtered knowledge graph embeddings are then combined with the hidden states from a BERT-based encoder and passed to the decoder.

Planning and constraint-guided generation Typical neural data-to-text models, which generate text from inputs like databases, often suffer from redundancy and lack factual faithfulness. Puduppully et al. (2019) proposed an alternative data-to-text approach where the input is a record table and the output is a natural language text. Their model explicitly separates content determination and content planning before passing the result to a neural generator for surface realization. The input is first encoded using a neural encoder. A content selection gate then determines relevant content using an attention mechanism over the table entries, followed by a sigmoid activation to determine which content is selected for further processing. Next, the content planning module decides what to say and in what order by generating a sequence of selected records using a pointer network. These plans are learned by aligning the summary text with table records. The resulting plan is then fed into a neural generator, which uses a standard encoder-decoder architecture to produce the output text.

Other data-to-text approaches include LogicNLG (Chen et al., 2020) and Symbolic Reasoning with Entity Scheduling (SORTIE; Zhao et al., 2023) which frame the task as logical data-to-text generation and aim to produce text that is logically consistent with the input data.

Lu et al. (2021) propose logic-guided, constraint-based generation that controls the decoding stage of neural text generation. It uses negative and positive constraints expressed as predicate logic formulas, which are converted into a penalty term and added to the decoding objective. This allows the model to generate fluent output while satisfying symbolic constraints, effectively guiding generation through inference-time decoding.

4.2 Vision

Graph-based methods In order to obtain agents which are able to proficiently understand the tem-

poral, relational and causal dynamics that go into performing everyday house-hold tasks, (Hazra et al., 2023) proposes a benchmark called Egocentric Task Verification (EgoTV), comprising a set of egocentric videos of daily life tasks, accompanied by a natural language description, as well as a novel Neuro-Symbolic Grounding (NSG) approach to counter the low performance exhibited by existing vision-language models on the EgoTV benchmark. The NSG architecture can convert a task description into a graph through its different components. This graph is then grounded in the video frames and the information represented by its nodes is aligned with the video. The NSG approach proposed by the authors indeed proved able to outperform state-of-the-art VLMs in capturing tasks' steps on both the EgoTV benchmark and on a dataset derived from the CrossTask dataset (Zhukov et al., 2019).

Huang et al. (2025) show interest in the understanding of spatio-temporal dynamics in videos as well. They introduce LASER, a neuro-symbolic framework that converts videos into graphs representing the properties and relations of entities at various time points. It then computes the alignment between these graphs and video captions that have been translated into formulas using an extended Linear Temporal Logic. The model is trained using weak supervision and displays enhanced performance compared to previous solutions in capturing relationships and dynamics in a range of video datasets with rich spatio-temporal specifications.

He et al. (2023) aim at applying scene graph generation to human-object interaction detection. They propose the unified model SG2HOI+ based on the Transformer architecture. The model is able to extract semantic-spatial features from images using a bounding box segmentation network, then generating scene graphs using information from said bounding boxes, and finally to convert the scene graphs into human-object interactions. SG2HOI+ was tested on a variety on benchmarks (Visual Genome, V-COCO, HICO-Det), and achieved better results than pre-existing methods.

Methods with programmatic descriptions

Gupta and Kembhavi (2023) X presents VisProg, a neuro-symbolic modular model that uses LLMs prompted in a few-shot manner to generate Python-like programs from image captions, questions and instructions. At each step, the programmes invoke one of the 20 modules currently supported (ranging from other LLMs to CLIP-like models and logic and arithmetic reasoning modules). VisProg performs at a high level on a range of of V&L tasks.

Hsu et al. (2023) introduce a modular architecture with neuro-symbolic components to solve 3D grounding tasks. It uses a language-to-code model to convert instructions in natural language asking to identify objects in pictures into symbolic programs. It then extracts object features and relations using an encoder, executes the program using the learned features and retrieves the target object.

Li et al. (2020) focus on jointly modeling camera poses, object locations and scene structures of naturalistic images presenting pronounced pattern regularities, treating the task as an inverse graphics problem, generating a graphic program from an input image, then reconstructing the picture and computing a loss between the reconstruction and the target.

Program-based neuro-symbolic approaches have also been applied to video related tasks. Kulal et al. (2021) introduces a framework designed to enhance human motion understanding in videos. It follows a hierarchical pipeline which first detects key points in videos, then produces both a concrete motion program, by assigning parameters to three motion primitives, and an abstract motion program, which generalizes over the concrete one by capturing higher-level repeated sequences and loops of primitives in the video.

5 Neuro-symbolic REG: Future directions

After reviewing approaches of neuro-symbolic integration methods in the area of REG and its closely related fields of Natural Language Generation and Vision (& Language), we will now discuss future directions to neuro-symbolic REG. In doing so, we will take into account the cognitively plausible properties of neuro-symbolic approaches – in particular, the combination of bottom-up, data-driven learning and structured reasoning about such data based on prior knowledge as two key aspects of human cognitive abilities and intelligent behaviour. In our discussion, we focus on the challenges in two different (but potentially overlapping) REG settings, i.e., interactive reference games set up as visual task-oriented dialogue, and reference generation under naturalistic real-world conditions.

Various neuro-symbolic approaches, previously discussed in Sections 3 and 4, seem fit to be adapted and applied to reference games, specifically *Cooperative* solutions (such as Chamorro-Martínez et al.,

2021; Huang et al., 2025; He et al., 2023), which are centered around the conversion of visual inputs into graphs representing their properties and relations, further used to solve tasks involving reasoning about those. Chamorro-Martínez et al. (2021) already applied such techniques, with an architecture able to generate referring expressions to objects in images employing fuzzy graphs.

Generating referring expressions in dialogical reference games, however, should consider the addressee and treat reference as a collaborative process (Clark and Wilkes-Gibbs, 1986), in which feedback is provided and the interaction history is available. While RSA-based models have a concept of a 'listener', it is an abstract one and not an actual addressee the model is interacting with and to whom the reference could be tailored. Adaptations to the addressee should happen on different levels of processing. Low level adaptations, such as interactive (lexical and syntactic) alignment (Pickering and Garrod, 2004) can be handled neurally during surface realization, whereas more strategic adaptations (Clark and Wilkes-Gibbs, 1986) could be the result of symbolic planning processes, which, however, could result from neural processing of the addressee's multimodal behaviors. If the addressee makes an error in, or is unable to resolve an initial reference, a model such as Chamorro-Martínez et al. (2021) could be adapted by implementing an iterative process, which retrieves the fuzzy graph previously produced and compares the target's node to that of the object wrongly selected by the addressee, in order to identify characteristics that were mistakenly chosen for or left out of the generated referring expression. The symbolic fuzzy graph is therefore a useful intermediate (and mediating) representation, that allows comparison of the speaker's and addressee's conceptualization with features of the target.

Nevertheless, more tightly integrated neuro-symbolic solutions such as those belonging to the *Compiled* class could be useful in modeling reference games, too. Methods including Logic Neural Networks (Riegel et al., 2020) and Logic Tensor Networks (Badreddine et al., 2022) could be used to extract features from images, encode these features as logical formulas, and then impose rules and constraints on them to guide the generation of referring expressions. Adressee's errors and feedback can be accounted for by updating the model weights or logic rules depending on whether the expression generated was precise enough for resolution.

A concern that can easily arise in REG tasks revolves around those cases where the exact category of an object that should be identified is not clear (Zarrieß and Schlangen, 2019). In such an eventuality, the knowledge injection methods discussed in Section 4.1 could potentially prove useful. In fact, through the use of knowledge graphs, it could be possible to provide REG models with knowledge bases granting them world knowledge, and thus informing them of more generic and overarching characteristics of objects that they might encounter in visual inputs, for instance common uses and functions, which could be used in indirect reference.

Cognitively oriented representations of the visual scenes in which referential targets are embedded also appear promising for generating referring expressions that are not only effective but also easy to understand. For example, Võ (2021) provides an in depth analysis of the rules and regularities of real world scenes, referred to as *Scene Grammars*. They include the notion of 'anchor objects', namely objects which are diagnostic of specific environments and serve as points of reference to identify other objects in a scene (e.g., the toilet in a bathroom). Inside visual scenes, objects tend to cluster around certain anchors, making it easier to identify them by restricting the domain of attention. The ability to take into consideration the pivotal role of anchors in object identification could be useful for REG models, as they would be able to abide to the natural way in which people parse visual scenes. *Cooperative* techniques based on graph structures, such as those presented in Section 4.2, could be optimized to recognize anchor nodes and subsequently use them to identify the target object in the scene, focusing on the relevant phrase and using information contained in it, such as the relationship the target has with the anchor, to construct referring expressions which can guide the listener's attention to the target in a way that is meaningful and familiar.

More generally, in vision and language tasks, such as referring expression generation, neuro-symbolic processing has great potential when providing both more autonomy as well as the ability for bidirectional information flow to components on all levels of processing. Neural vision components could implement theories of visual attention that respond to saliency and/or other features (e.g., Gestalts) from the visual scene such that they do not provide an exhaustive representation of the scene but are already selective or operate on a different

level of abstraction and thereby influence object naming or attribute selection when generating references with IA-like algorithms. Conversely, attention and visual processing could also be guided top down through symbolic information that is grounded in an interlocutor's utterance (such as a clarification), the broader interaction history, or the speaker agent's goal. Following theories of 'ecological perception' (Gibson, 1979), this could afford a neural re-conceptualization of objects in the scene, possibly yielding completely different features and object description that fit the speaker's need at a specific moment in a reference game.

6 Conclusion

Neuro-symbolic approaches are gaining considerable interest in computational linguistics and NLP, as they allow to integrate the complementary characteristics of symbolic and neural processing, potentially leading to strong and adaptive, but also transparent and cognitively plausible systems. In this paper, we reviewed existing neuro-symbolic approaches in REG and discussed possible future directions, drawing on related research areas such as NLG and vision. As an inherently multimodal task with defined pragmatic objectives, REG opens up many possibilities for linking these paradigms at different levels, opening up exciting possibilities for further research.

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project no. 512393437, project B02.

References

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, TX, USA.

Sebastian Bader and Pascal Hitzler. 2005. Dimensions of neural-symbolic integration – A structured survey. In Sergei Artemov, Howard Barringer, Artur d'Avila Garcez, Luis C. Lamb, and John Woods, editors, *We Will Show Them: Essays in Honour of Dov Gabbay*, pages 167–194. King's College Publications.

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic tensor networks. *Artificial Intelligence*, 303:103649.

Anja Belz, Eric Kow, Jette Viethen, and Albert Gatt. 2010. Generating referring expressions in context: The GREC task evaluation challenges. In Emiel Krahmer and Theune Mariët, editors, *Empirical Methods in Natural Language Generation*, pages 294–327. Springer, Berlin, Germany.

Lior Bracha, Eitan Shaar, Aviv Shamsian, Ethan Fetaya, and Gal Chechik. 2023. DisCLIP: Open-vocabulary referring expression generation. In *Proceedings of the 34th British Machine Vision Conference*, Aberdeen, UK.

Andrea Cadeddu, Alessandro Chessa, Vincenzo De Leo, Gianni Fenu, Enrico Motta, Francesco Osborne, Diego Reforgiato Recupero, Angelo Salatino, and Luca Secchi. 2024. A comparative analysis of knowledge injection strategies for large language models in the scholarly domain. *Engineering Applications of Artificial Intelligence*, 133:108166.

Jesús Chamorro-Martínez, Nicolás Marín, Míriam Mengíbar-Rodríguez, Gustavo Rivas-Gervilla, and Daniel Sánchez. 2021. Referring expression generation from images via deep learning object extraction and fuzzy graphs. In *Proceedings of the 2021 IEEE International Conference on Fuzzy Systems*, pages 1–6, Luxembourg.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

Reuben Cohn-Gordon, Noah D. Goodman, and Christopher Potts. 2018. Pragmatically informative image captioning with character-level inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 439–443, New Orleans, Louisiana.

Robert Dale. 1989. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 68–75, Vancouver, Canada.

Robert Dale and Nicholas Haddock. 1991. Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252–265.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean Maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Michael C. Frank, Andrés Gómez Emilsson, Benjamin Peloquin, Noah D. Goodman, and Christopher Potts. 2016. Rational speech act models of pragmatic reasoning in reference games. *Preprint*, OSF:f9y6b.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336:998.

Artur d'Avila Garcez and Luís C. Lamb. 2023. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406.

Albert Gatt and Kees van Deemter. 2007. Lexical choice and conceptual perspective in the generation of plural referring expressions. *Journal of Logic, Language and Information*, 16(4):423–443.

James J. Gibson. 1979. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA.

Herbert Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics 3: Speech Acts*, pages 41–58. Academic Press, New York, NY, USA.

Danfeng Guo, Sanchit Agarwal, Arpit Gupta, Jiun-Yu Kao, Emre Barut, Tagyoung Chung, Jing Huang, and Mohit Bansal. 2024. Prompting vision-language models for aspect-controlled generation of referring expressions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2793–2807, Mexico City, Mexico.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, Vancouver, Canada.

Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. 2024. Is neuro-symbolic AI meeting its promises in natural language processing? A structured review. *Semantic Web*, 15(4):1265–1306.

Rishi Hazra, Brian Chen, Akshara Rai, Nitin Kamra, and Ruta Desai. 2023. EgoTV: Egocentric task verification from natural language task descriptions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15371–15383, Paris, France.

Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. 2023. Toward a unified transformer-based framework for scene graph generation and human-object interaction detection. *IEEE Transactions on Image Processing*, 32:6274–6288.

Helmut Horacek. 1997. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 206–213, Madrid, Spain.

Helmut Horacek. 2004. On referring to sets of objects naturally. In *Proceedings of the 3rd International Conference on Natural Language Generation*, pages 70–79, Brockenhurst, UK. Springer.

Joy Hsu, Jiayuan Mao, and Jiajun Wu. 2023. NS3D: Neuro-symbolic grounding of 3d objects and relations. In *In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2614–2623, Vancouver, Canada.

Jiani Huang, Ziyang Li, Mayur Naik, and Ser-Nam Lim. 2025. LASER: A neuro-symbolic framework for learning spatial-temporal scene graphs with weak supervision. In *Procedings of the 13th International Conference on Learning Representations*, Singapore.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Simeon Junker and Sina Zarrieß. 2024. Resilience through scene context in visual referring expression generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 344–357, Tokyo, Japan.

Henry Kautz. 2022. The third AI summer: AAAI Robert S. Engelmore memorial lecture. *AI Magazine*, 43(1):105–125.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar.

John D. Kelleher and Geert-Jan M. Kruijff. 2006. Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1041–1048, Sydney, Australia.

Jungjun Kim, Hanbin Ko, and Jialin Wu. 2020. CoNAN: A complementary neighboring-based attention network for referring expression generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1952–1962, Barcelona, Spain (Online).

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. 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 2284–2293, Minneapolis, MN, USA.

Emiel Krahmer and Mariet Theune. 2002. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, pages 223–264. CSLI, Stanford, CA, USA.

Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

Sumith Kulal, Jiayuan Mao, Alex Aiken, and Jiajun Wu. 2021. Hierarchical motion understanding via motion programs. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6564–6572, Los Alamitos, CA, USA.

Xiangyang Li and Shuqiang Jiang. 2018. Bundled object context for referring expressions. *IEEE Transactions on Multimedia*, 20(10):2749–2760.

Yikai Li, Jiayuan Mao, Xiuming Zhang, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. 2020. Perspective plane program induction from a single image. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4433–4442, Seattle, WA, USA.

Yaoyuan Liang, Zhuojun Cai, Jian Xu, Guanbo Huang, Yiran Wang, Xiao Liang, Jiahao Liu, Ziran Li, Jingang Wang, and Shao-Lun Huang. 2024. Unleashing region understanding in intermediate layers for MLLM-based referring expression generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 120578–120601.

Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. 2017. Referring expression generation and comprehension via attributes. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy.

Jingyu Liu, Wei Wang, Liang Wang, and Ming-Hsuan Yang. 2020. Attribute-guided attention for referring expression generation and comprehension. *IEEE Transactions on Image Processing*, 29:5244–5258.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2023. Unified-IO: A unified model for vision, language, and multi-modal tasks. In *Proceedings of the 11th International Conference on Learning Representations*, Kigali, Rwanda.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. NeuroLogic decoding: (Un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online.

Ruotian Luo and Gregory Shakhnarovich. 2017. Comprehension-guided referring expressions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3125–3134, Honolulu, HI, USA.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. DeepProbLog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, volume 31, pages 1–11, Montréal, Canada.

Junhua Mao, J. Huang, A. Toshev, Oana-Maria Camburu, A. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, Las Vegas, NV, USA.

Bill McDowell and Noah D. Goodman. 2019. Learning from omission. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 619–628, Florence, Italy.

Nikolaos Panagiaris, Emma Hart, and Dimitra Gkatzia. 2020. Improving the naturalness and diversity of referring expression generation models using minimum risk training. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 41–51, Dublin, Ireland.

Nikolaos Panagiaris, Emma Hart, and Dimitra Gkatzia. 2021. Generating unambiguous and diverse referring expressions. *Computer Speech & Language*, 68:101184.

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Synergizing machine learning & symbolic methods: A survey on hybrid approaches to natural language processing. *Expert Systems with Applications*, 251:124097.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conference and 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 6908–6915, Honolulu, HI, USA.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK.

Ryan Riegel, Alexander Gray, Francois Luus, Naweed Khan, Ndivhuwo Makondo, Ismail Yunus Akhalwaya, Haifeng Qian, Ronald Fagin, Francisco Barahona, Udit Sharma, Shajith Ikbal, Hima Karanam, Sumit Neelam, Ankita Likhyani, and Santosh Srivastava. 2020. Logical neural networks. *Preprint*, arXiv:2006.13155.

Simeon Schüz, Ting Han, and Sina Zarrieß. 2021. Diversity as a by-product: Goal-oriented language generation leads to linguistic variation. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 411–422, Singapore and Online.

Simeon Schüz and Sina Zarrieß. 2021. Decoupling pragmatics: Discriminative decoding for referring expression generation. In *Proceedings of the Reasoning and Interaction Conference (ReInAct 2021)*, pages 47–52, Gothenburg, Sweden.

Simeon Schüz, Albert Gatt, and Sina Zarrieß. 2023. Rethinking symbolic and visual context in referring expression generation. *Frontiers in Artificial Intelligence*, 6:1067125.

Advaith Siddharthan and Ann Copestake. 2004. Generating referring expressions in open domains. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 407–414, Barcelona, Spain.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Matthew Stone and Bonnie Webber. 1998. Textual economy through close coupling of syntax and semantics. In *Proceedings of the 9th International Workshop on Natural Language Generation*, pages 178–187, Niagara-on-the-Lake, Canada.

Mengyang Sun, Wei Suo, Peng Wang, Yanning Zhang, and Qi Wu. 2022. A proposal-free one-stage framework for referring expression comprehension and generation via dense cross-attention. *IEEE Transactions on Multimedia*, 25:2446–2458.

Mikihiro Tanaka, Takayuki Itamochi, Kenichi Narioka, Ikuro Sato, Yoshitaka Ushiku, and Tatsuya Harada. 2019. Generating easy-to-understand referring expressions for target identifications. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5793–5802, Seoul, Korea.

Polina Tsvilodub, Michael Franke, and Fausto Carcassi. 2024. Cognitive modeling with scaffolded LLMs: A case study of referential expression generation. In *ICML 2024 Workshop on LLMs and Cognition*, Vienna, Austria.

Leslie G. Valiant. 2003. Three problems in computer science. *Journal of the ACM*, 50(1):96–99.

Kees van Deemter. 2016. *Computational Models of Referring: A Study in Cognitive Science*. The MIT Press, Cambridge, MA, USA.

Kees van Deemter, Albert Gatt, Roger P.G. van Gompel, and Emiel Krahmer. 2012. Toward a computational psycholinguistics of reference production. *Topics in Cognitive Science*, 4(2):166–183.

Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the 4th International Natural Language Generation Conference*, pages 130–132, Sydney, Australia.

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1070–1079, Honolulu, HI, USA.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA.

Melissa Le-Hoa Võ. 2021. The meaning and structure of scenes. *Vision Research*, 181:10–20.

Guan Wang, Weihua Li, Edmund Lai, and Jianhua Jiang. 2022a. KATSum: Knowledge-aware abstractive text summarization. In *Proceedings of the 2022 Principle and Practice of Data and Knowledge Acquisition Workshop PKAW*, Shanghai, China.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022b. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23318–23340.

Julia White, Jesse Mu, and Noah D. Goodman. 2020. Learning to refer informatively by amortizing pragmatic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 42, Virtual.

Bram Willemsen and Gabriel Skantze. 2024. Referring expression generation in visually grounded dialogue with discourse-aware comprehension guiding. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 453–469, Tokyo, Japan.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, Seattle, WA, USA.

Fulong Ye, Yuxing Long, Fangxiang Feng, and Xiaojie Wang. 2023. Whether you can locate or not? Interactive referring expression generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4697–4706, Ottawa, ON, Canada.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. FERRET: Refer and ground anything anywhere at any granularity. In *The 12th International Conference on Learning Representations*, Vienna, Austria.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In *Computer Vision – ECCV 2016*, pages 69–85, Cham, Switzerland. Springer.

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L. Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3521–3529, Honolulu, HI, USA.

Sina Zarrieß and David Schlangen. 2016. Easy things first: Installments improve referring expression generation for objects in photographs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 610–620, Berlin, Germany.

Sina Zarrieß and David Schlangen. 2018. Decoding strategies for neural referring expression generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 503–512, Tilburg, The Netherlands.

Sina Zarrieß and David Schlangen. 2019. Know what you don't know: Modeling a pragmatic speaker that refers to objects of unknown categories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 654–659, Florence, Italy.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2031–2043, Online.

Xueliang Zhao, Tingchen Fu, Lemao Liu, Lingpeng Kong, Shuming Shi, and Rui Yan. 2023. SORTIE: Dependency-aware symbolic reasoning for logical data-to-text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11247–11266, Toronto, Canada.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. 2019. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3532–3540, Long Beach, CA, USA.

Extracting a Prototypical Argumentative Pattern in Financial Q&As

Giulia D'Agostino
Università della Svizzera italiana
Switzerland
giulia.dagostino@usi.ch

Michiel van der Meer
Universiteit Leiden
The Netherlands
m.t.van.der.meer@liacs.leidenuniv.nl

Chris Reed
University of Dundee
Scotland
c.a.reed@dundee.ac.uk

Abstract

Argumentative patterns are recurrent strategies adopted to pursue a definite communicative goal in a discussion. For instance, in Q&A exchanges during financial conference calls, a pattern called Request of Confirmation of Inference (ROCOI) helps streamline conversations by requesting explicit verification of inferences drawn from a statement. Our work presents two ROCOI extraction approaches from interrogative units: sequence labeling and text-to-text generation. We experiment with multiple models for each task formulation to explore which models can effectively and robustly perform pattern extraction. Results indicate that machine-based ROCOI extraction is an achievable task, though variation among metrics that are designed for different evaluation dimensions makes obtaining a clear picture difficult. We find that overall, ROCOI extraction is performed best via sequence labeling, though with ample room for improvement. We encourage future work to extend the study to new argumentative patterns.

1 Introduction

An argumentative pattern is a recurrent and identifiable structure with a specific function in an argumentative discussion. Such a pattern offers valuable insights into the reasoning processes and dialectical strategies employed by interlocutors in argumentative discourse.

Extracting argumentative patterns from natural discourse presents a significant challenge in the field of Argument mining (AM) (Lawrence and Reed, 2019). Typically, AM involves three stages: (1) the identification, segmentation, and classification of argumentative discourse units (ADUs) (Ghosh et al., 2014), (2) the characterization of the relations between ADUs (Peldszus and Stede, 2013), and (3) the identification of argument schemes, which denote implicit and explicit inferential relations within and across ADUs (Macagno and Walton, 2014). This area of research is often challenged by the idiosyncrasies of spoken language. For instance, in Earnings Conference Call (ECC) Q&A sessions, argumentative content is often embedded in complex statements aimed at maximizing information content while minimizing exchanges (Keith and Stent, 2019). Instead of employing a typical end-to-end AM pipeline, leveraging linguistic patterns that are clearly identifiable as part of argument schemes could be useful for locating argumentative moves, unraveling the complexities in such dialogues.

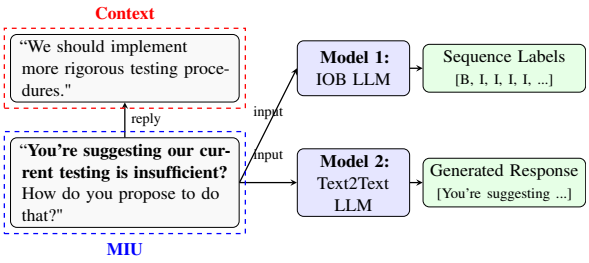


Figure 1: Example ROCOI and the two extraction approaches.

In this paper, we present a novel task and approach to the extraction of a prototypical argumentative pattern called the **Request Of Confirmation Of Inference** (ROCOI). Our work focuses on this argumentative pattern that emerges in questions, which presents some easily identifiable surface elements that complement the underlying argumentative function.

The ROCOI pattern is a structure that signals the presence of a reasoning process, of which the interrogative instance represents the conclusion (or, more accurately, the request for confirmation thereof). By carefully bridging lexical and syntactic recurrent features with the pragmatic role that such a pattern plays, ROCOIs constitute a unique pattern whereby we shed light on the reasoning process behind strategic inquiry. In the extraction

Proceedings of the 2025 CLASP Conference on Language models And RePresentations (LARP), pages 51–64

process, we move beyond the analysis of entire discourse units, instead allowing us to localize ROCOIs inside dialogues. This approach allows us to maintain precise control over pattern detection while dealing with the inherent complexity of argumentative texts. In this sense, the current approach moves beyond pattern identification–a task already tackled by D'Agostino and Rocci (2024)–towards pattern extraction.

We specifically focus on the ROCOI pattern as it represents an ideal proto-pattern for exploring how well NLP methods can extract argumentative patterns from text. These patterns exhibit characteristics that make them readily identifiable by trained human annotators, including their interrogative nature on an inferential conclusion, and explicit marking of prior reasoning. This clarity provides an excellent starting point for developing and evaluating automated extraction methods. At the same time, such characteristics are neither regular nor exclusive to this task, so they must be paired with recognition of the argumentative reasoning that the patterns reside in Since LLMs may not behave like human annotators in argumentative reasoning (de Wynter and Yuan, 2024), with this study, we probe the limits of pragmatic pattern recognition by means of surface elements.

The selection of our domain of application is mainly utilitarian: ECCs are strategic exchanges that constrain discourse in a way that makes the ROCOI pattern relatively frequent and prominent–an optimal environment for an exploratory study.

Our experiments encompass two task formulations, comparing a classification (sequence labeling) and generation (text-to-text extractive generation) paradigm. Comparing these two approaches allows us to bridge traditional boundary-marking techniques (Eger et al., 2017; Kuribayashi et al., 2019; Bao et al., 2021) with state-of-the-art language modeling approaches (Raffel et al., 2020; Gorur et al., 2024).

This work represents a crucial step toward the broader goal of comprehensive argumentative analysis, laying the groundwork for future exploration of more complex patterns, as well as the incorporation of contextual features in detecting argumentative patterns. Furthermore, our models can support humans in locating argumentation in financial contexts (van der Meer et al., 2024), with potential applications in areas such as investor relations, corporate communication, and financial analysis.

2 Related Work

2.1 The Request of Confirmation of Inference: An argumentative pattern in ECCs

The ROCOI (previously introduced and qualitatively studied by Rocci and Raimondo, 2018) is an argumentative pattern in ECCs that is originated in question units. It is *relevant* to the discussion in the sense that it creates an argumentative confrontation (van Eemeren and Grootendorst, 2004).

A ROCOI is an assertive question, i.e., in which a stance is asserted by the questioner. As a consequence, when it is formulated directly, a ROCOI is a closed question. Moreover, ROCOIs make explicit by lexical means the fact that the stance asserted is the result of an inferential process, the conclusion of which is expected to be (dis)confirmed by the interlocutor. This results in the ROCOI being a challenging question, regardless of the degree of semantic indirectness of its formulation.

Example 1 shows some ROCOIs; underlined, the lexical elements that indicate the inferential process, which constitutes the keystone of the pattern.

(1) a. Does that mean that customers are reluctant to term out these sort of prices?

b. Should we think of the capital commitment has a hard cap now.

c. Is it fair to say that you've maxed out on what was pre-approved at the AGM and that any incremental issue from here would require AGM approval?

Previous studies on the ROCOI (Rocci and Raimondo, 2018; D'Agostino and Rocci, 2024) identify subcategories of the pattern. This article considers the class that D'Agostino and Rocci (2024) call Type 1, that is, ROCOIs in which the inferential conclusion–of which the questioner asks confirmation–is part of the interrogative unit, as shown in all questions of Example 1. On the other hand, Type 2 ROCOIs correspond to patterns in which the questioner's inference is not explicitly part of the interrogative sentence, such as in the following example (ROCOI in italics): "And looking at the take rate in the fourth quarter, might have slowed down a little bit. *So just – is that true?*". The reason behind the choice of experimenting with Type 1 only is twofold: on the one hand, Type 1 ROCOIs are more compact, in the sense that the conclusion and the question pertain to the same

unit, and are therefore more easily identifiable; on the other hand, they are the most frequent ones.

Easiness in identification should not be mistaken with triviality of the task, and so the retrieval of this pattern cannot be simply achieved via rule-based search of key-phrases such as those emphasized in Example 1. The reason is twofold. The typical lexical signals that indicate the presence of a ROCOI belong to the domain of knowledge management (epistemicity and evidentiality) (Musi and Rocci, 2017; Miecznikowski, 2020; Lucchini et al., 2024); however, while not all ROCOIs display them, these indicators also have a much wider scope of application than introducing this question type. Conversely, the retrieval of such key-phrases does not ensure the extraction of the entire pattern, as it does not provide any indication about its extension–which is not predefined.

3 Method

We outline the dataset, task formulation, and evaluation setup for the ROCOI extraction approaches.

3.1 Dataset

Our work focuses on a dataset that comprises 60 Earnings Conference Calls (ECCs) between 2020–2023 for companies Airbnb (ABNB), British Petroleum (BP), Credit Suisse (CS), Door Dash (DASH), Hasbro (HAS), Shell (SHEL), Exxon Mobil (XOM) and Zillow (Z), for a total of 1377 question units. Manual annotation identified 180 question units featuring ROCOIs, in total containing 193 unique ROCOI patterns. Of these, 134 were Type 1 ROCOIs, and thus represent the final corpus for this study.

The annotation was first carried out by trained student assistants. Annotators were MA students selected on the basis of their joint background in linguistics/languages and financial communication. Whereas financial literacy supports domain knowledge and text comprehension, higher impact in annotation quality is credited to linguistic awareness: ROCOIs are, in fact, a linguistic phenomenon that happens to be frequent in this context, but whose form is not influenced by the content.

Each document was analyzed by two to four annotators. The agreement on the request type labeling task (for which the ROCOI is one out of eight possible values) is $\alpha = 0.79$ (Krippendorff, 1970)[1]; agreement on ROCOI span length is $\Gamma = 0.52$ (Mathet et al., 2015). Two PhD students subsequently curated the annotation until reaching a shared gold standard. This was followed by an additional round of dictionary-based search of (potential) remaining instances, performed by the first contributor of the current paper. Further information about the annotation guidelines for request types is provided in Lucchini and D'Agostino (2023, p. 15-19); ROCOI type classification is borrowed from D'Agostino and Rocci (2024). In total, 18% of tokens in the dataset are part of a ROCOI, whereas 82% of tokens are non-ROCOI tokens. At the present stage and to the best of our knowledge, this is the most extensive collection of manually annotated ROCOIs.

3.2 Task formulation

We compare two task formulations for ROCOI extraction: (1) sequence labeling and (2) text generation, applied to interrogative units that were previously identified as exhibiting the pattern in question. These two tasks allow us to compare the results obtained from applying a *classification* and a *generation* paradigm. Classification, where we mark the boundaries between the presence and absence of a ROCOI, represents the standard method of identifying a substructure. However, such an approach usually requires ample training data. In contrast, text-to-text extractive generation, which involves generating the part of the input text that contains the ROCOI pattern, is similar to more recent state-of-the-art LLMs. We aim to investigate which approach works better given our relatively small dataset. We describe each task formulation separately and provide details about hyperparameters and training settings for all models in Appendix A. The results of both these experiments are compared against an LLM-generated baseline (decoder-only architecture), obtained by prompting the GPT-4o API. Seven-shot in-context learning was adopted as prompting technique for sequence labeling, five-shot for text-to-text generation.[2]

(1) Sequence labeling Sequence labeling for ROCOI extraction is formulated as a task whereby we mark the boundaries between the presence and ab-

[1]Disagreements relate to the selection of a different request type, mostly biased by the *content* of the inference: a ROCOI may be tagged as a request for opinion if, for instance, the inference is about an opinion that the management may hold. No trends were found upon disagreement analysis.

[2]Models GPT-4o, GPT-4o-mini, and GPT-4.1 were compared across 0-, 3-, 5-, and 7-shot contexts; reported as "LLM baseline" is the combination that performed best on average across metrics for the task.

sence of the pattern. Each token is classified as whether pertaining to the sequence (tags "B" and "I"), or not (tag "O"). Such a format represents the standard method of identifying a substructure in a text–for instance, for Named Entity Recognition (NER). Unlike traditional NER, we only consider one type of pattern and thus do not need to specify the class to which the tags pertain.

We experiment with 5 open-source models in total; three of those are encoder-only models:

TinyBERT The smallest model to gauge task complexity. If the smallest model can learn it well, we do not need to train a more capable model (Jiao et al., 2020).

Vanilla BERT Since it is commonly used as a baseline (Devlin et al., 2019).

SpanBERT As a version of BERT that is optimized to represent spans of text, since ROCOIs are often single contiguous spans (Joshi et al., 2020).

In addition, we also experiment with two encoder-decoder models:

T5 Strong empirical results indicate that this model may be used across contexts and tasks (Raffel et al., 2020).

FlanT5 Updated version of T5 that includes a wider array of tasks, the model may generalize better to unseen tasks (Longpre et al., 2023).

(2) Text-to-text generation For this task, the pattern is considered a substring of the question unit given as an input; hence, the output corresponds to a *verbatim* generation of a portion of the wider unit (similar to the use of the text-to-text architecture already intended by Raffel et al. (2020)). Therefore, particular attention must be devoted to the quality of the generation and, specifically, that the fine-tuned model reports an exact portion of the original text (and not, for instance, a summarization of it) and learns that a pattern is a continuous sequence within the text.

This portion of the study is carried out on two text-to-text model families:

BART serves as the encoder-decoder counterpart to our BERT baseline for sequence labeling. We use the base and large varieties (Lewis et al., 2020) to further investigate the impact of model size.

T5 in the small, base, and large varieties, again to see whether a more versatile text-to-text training procedure benefits performance (Raffel et al., 2020).

3.3 Evaluation

We outline how we evaluate models on each task formulation.

3.3.1 Sequence labeling

We initially aimed to adopt a similar evaluation approach as Named Entity Recognition (NER), as it shares the IOB tagging setup (Li et al., 2020). Performance in NER and similar tasks is traditionally evaluated at the **token level** (Tjong Kim Sang and Buchholz, 2000). However, tagging is typically performed (a) on short sequences, (b) in multiclass classification, and (c) featuring multiple units in a text; none of these characteristics strictly hold for ROCOIs. Even in the NER extraction domain, however, there has been a propensity towards evaluation at the full entity level, especially if the prediction is aimed at downstream tasks (Segura-Bedmar et al., 2013). Since ROCOIs are long and complex spans of text with potentially variable boundaries, we additionally adopt **span-level evaluation** and compare it to individual token-level evaluation.

Token-level evaluation At the token level, we first provide an overview of the accuracy in the prediction by individual tags ('O', 'I', 'B'). Then we aggregate the tags and provide a measure of precision, recall, and F1 score, alongside the calculation of token-based Krippendorff's α (Krippendorff, 1970).

Span-level evaluation To evaluate the entire span over which the ROCOI develops and not only the individual tokens that constitute it, we make use of the ROUGE-L metric, to determine the longest matching string, as well as the Gamma (Γ) method for inter-annotator agreement measure and alignment (Mathet et al., 2015)[3] in a basic, one-label, positional dissimilarity detection configuration.

3.3.2 Text-to-text generation

For the text-to-text generation evaluation, we use various metrics to investigate the quality of the extracted pattern. Each model is evaluated according to six metrics, clustered into three classes, each of which corresponds to a different way of interpreting the nature of the task: *syntactic* (pattern matching), *semantic* (embedding similarity), or *annotation* (inter-annotator agreement). The rationale behind such a three-fold choice lies in the nature of generative models: on the one hand, they tend to

[3]Taken from the Python library `pygamma-agreement` (`https://github.com/bootphon/pygamma-agreement`)

be too creative despite being nudged to extract verbatim text. This would not be captured by semantic metrics but is counterbalanced by syntactic metrics. On the other hand, syntactic evaluation cannot capture whether some slightly shifted boundary still correctly identifies the core of the pattern–which can however be reintegrated into the equation to some extent by the use of semantic similarity (although not entirely, since such metrics are not specialized in ROCOI *core meaning* detection, similar to sequence labeling). Inter-annotator agreement metris works as a sanity check that decidedly signals the presence of ill-formed sequences in generated patterns.

Syntactic evaluation In this view, the extraction performance is evaluated in terms of string matching. The first naïve evaluation that establishes the baseline consists of checking whether the pattern is present in the extracted string. We call this evaluation "pattern matching" and its most obvious flaws are that (a) over-extraction to the point of reporting the entire original string is a hit and (b) even slight under-extraction is a complete miss. The three possible values are 'full match' (if the retrieved string contains exactly the correct pattern), 'partial match' (if the retrieved string contains at least the full correct pattern), and 'no match' otherwise; reported are the frequency distributions across the three classes. This is paired with a more refined version of such an evaluation, that is, the calculation of the ROUGE score (Lin, 2004); specifically the ROUGE-L metric, which identifies the longest co-occurring sequence.

Semantic evaluation In this case, what is evaluated is the semantic distance between the predicted and the actual pattern. This is achieved by (1) calculating a simple Euclidean distance between the embedding representation of the patterns and (2) applying some well-established evaluation methods that are typically used for text generation and summarization: notably (a) BERTScore (Zhang et al., 2020) and (b) Sentence-BERT (SBERT) (Reimers and Gurevych, 2019).[4]

Annotation agreement evaluation The true pattern can be considered a gold standard annotation and the extracted pattern a machine-generated annotation; in this perspective, the two are compared with a tool designed to capture the inter-annotator agreement and the dissimilarity in span boundaries.

[4]SBERT in its base configuration measures cosine similarity.

	Accuracy		
Model	O	I	B
BERT (base)	<u>0.93</u>	<u>0.61</u>	**0.70**
TinyBERT	0.92	0.39	0.40
SpanBERT	0.89	<u>0.61</u>	0.60
T5 (base)	**0.95**	**0.67**	<u>0.65</u>
FlanT5 (base)	0.92	**0.67**	**0.70**
GPT-4o	*0.83*	*0.40*	*0.05*

Table 1: Sequence labeling accuracy by tag. The best models are shown in bold, second best underlined. The LLM baseline is in italic.

In particular, we use the Gamma (Γ) method for inter-annotator agreement measure and alignment (Mathet et al., 2015). The metric cannot compute on instances in which the extracted pattern is not a lexical match to a substring of the input text, and thus tells us that the generated string is ill-formed.

4 Results and discussion

We describe our results after training the models on the two tasks: sequence labeling and text-to-text generation respectively.

4.1 Sequence labeling

Table 1 reports the accuracy values by individual tag. As reported in Section 3.1, 82% of tokens in the dataset are non-ROCOI elements; these are identified by 'O' tags. Therefore, since they represent the most frequent type, as expected 'O' tokens reach a higher accuracy across models. On the contrary, 'B'-type tokens understandably are the least frequent ones in the corpus but its accuracy levels are not far from that of 'I' tokens overall–if not better. It is worth noticing that SpanBERT appears to be performing badly despite being optimized for encoding contiguous spans of texts. It achieves the lowest accuracy on the 'O' tag, indicating it most strongly mislocates ROCOI patterns in the text. The LLM baseline confirms the accuracy trends but uniformly scores lower than any other model. At this stage, the best performing models seem to be the two belonging to the T5 family (both best in two out of three accuracy values), followed by vanilla BERT (second best in two out of three accuracy values).

Further classification results aggregated over the three tag categories are displayed in Table 2, both at the token level (former four columns) and span level (latter two columns). Token-level evaluation

appears to favor FlanT5, which achieves the highest results in three out of four metrics and is second best in the remaining one. Surprisingly, SpanBERT performs below par in full span detection, according to span-level evaluation results, which are instead dominated again by T5 (ROUGE-L = 0.90) and FlanT5 ($\Gamma = 0.63$). To conclude, the performance exhibited by T5, FlanT5, and TinyBERT on sequence labeling at the span level compares with or exceeds human agreement (i.e, $\Gamma \geq 0.52$). This indicates that we may use automatic ROCOI extraction for machine annotation for new samples in the future. However, the machine annotations fail in a way that is not captured by this metric, or disagree with human annotators in novel ways. Hence, we set out to further understand the limitations of the automatic ROCOI extraction approach in Section 5.1.

The LLM baseline confirms weak over both token- and span-level labeling, displaying for most metrics below average to nearly zero agreement. Tag sequences appear to be well formed, but not labeling the pattern correctly; moreover, the returned sequence is shorter than the reference in 98% of cases. This indicates that the model is unfit for the job, even though sequence labeling is a generation task in the linguistic domain–which is supposedly the type of task at which these models excel.

4.2 Text-to-text generation

We present the evaluation results sorted by evaluation approach type (*syntactic*, *semantic*, *annotation*), each of which is presented in a dedicated table.

Table 3 reports *syntactic* evaluation. For both evaluation methods, the two BART models appear to be by far the best-performing ones, particularly the *large* configuration–with best results across all metrics. *Semantic* measures are reported in Table 4. The baseline metric represented by raw Euclidean distance between the true and predicted pattern favors BART models; moreover, both SBERT and BERTScore, again identify BART-large as the best-performing model, reaching $F1 = 0.94$. Similar outcomes are shown in Table 5, which displays surprisingly bad results for the T5 models on the *inter-annotator agreement* metrics. This will be appropriately discussed in Section 5.2.

Different metrics capture different aspects of the ROCOI extraction task in a text-to-text generation setup, For instance, syntactic pattern matching informs us of the capability to lexically overlap with the ground truth patterns, while semantic evaluation allows us to observe how well the model captures the underlying meaning and intent of the ROCOI spans. We observe that BART models achieve good performance along all three dimensions for this task.

It is worth noting that for the text-to-text generation task, in contrast with what previously observed for sequence labeling, the tested models are often outperformed by the LLM baseline; this is especially evident in the semantic and annotation agreement metrics. This practically means that pattern boundaries detection may not be extremely accurate in the majority of cases, but the core content of the reference sequence is often included in the generated one. The good level of annotation agreement, moreover, ensures that the generated text is sufficiently well-formed with reference to the original pattern string.

5 Error analysis

In addition to our previous results, we present a qualitative analysis of the predicted patterns. Specifically, we observe the onset point and length of all extracted patterns in the test set to identify whether models tend to make consistent mistakes.

Further, we also present an overview of the distribution of ill-formed sequences in the prediction. In the sequence labeling task, this corresponds to cases in which a sequence onset is not correctly followed by the next element in the sequence: a 'B' tag is immediately followed by an 'O' (not possible in a well-formed ROCOI). In the text generation task, this corresponds to token sequences that are inconsistent with the original text.

While the results here summarize the findings, Tables 9 and 10 for sequence labeling and text generation respectively–available in Appendix B–report by row the measures over each instance in the test set.

5.1 Sequence labeling

We compare performance between T5 and FlanT5.

As for T5, perfect alignment with the start of the pattern occurs in 60% of cases, while the majority of predicted patterns appear to be shorter than expected (55%). Worth noting is the near-perfect acquisition of the IOB-tagging rules, which is reflected in a single instance of ill-formed sequence.

As for FlanT5, the right starting point is detected in 70% of cases; extraction of exact right length

Model	Token-level Precision	Recall	F1	α	Span-level ROUGE-L	Γ
BERT (base)	0.22	0.25	0.23	0.58	0.87	0.49
TinyBERT	0.09	0.05	0.06	0.37	0.82	0.60
SpanBERT	0.27	**0.30**	0.29	0.51	0.83	0.47
T5 (base)	0.17	0.15	0.16	**0.67**	**0.90**	0.56
FlanT5 (base)	**0.32**	**0.30**	**0.31**	0.61	0.87	**0.63**
GPT-4o	*0.03*	*0.02*	*0.02*	*0.28*	*0.77*	*0.39*

Table 2: Additional results for the sequence labeling approaches. The best models are shown in bold, second best underlined. The LLM baseline is in italic.

Model	Pattern matching Full match	Partial match	No match	ROUGE-L
BART (base)	**0.20**	0.50	0.30	0.63
BART (large)	**0.20**	**0.60**	**0.20**	**0.67**
T5 (small)	0.00	0.45	0.55	0.43
T5 (base)	0.15	0.50	0.35	0.54
T5 (large)	0.00	0.15	0.85	0.31
GPT-4o	***0.40***	*0.44*	***0.16***	***0.72***

Table 3: Syntactic evaluation for text-to-text generation. For pattern matching, results must be read as "the higher the better" for full and partial match, and "the lower the better" for no match. The best models are shown in bold, second best underlined. The LLM baseline is in italic; additionally, baseline results are in bold if their value is equal or better than the best result.

spurts to 30%. However, 100% of predicted patterns contain 1 to 4 ill-formed sequences. If the extraction process was integrated in a pipeline, this would easily result in error propagation.

Following, a test instance misclassified by both models (ROCOI pattern in italics): "And secondly, on U.S. gas, you're very well-positioned with I believe pretty much fully hedged production for this year, but *I'm wondering if at $2 per MCF gas, you're actually starting to see the opportunity to perhaps take away some of the rigs* and refocus them in the Permian where you keep strongly growing the activity. Thank you."

In this example, FlanT5 recognizes three starting points (underlined the tokens corresponding to a 'B' tag in the predicted sequence) and one well-formed sequence roughly corresponding to the true pattern (in bold the tokens corresponding to 'I' tags): "And secondly, on U.S. gas, you're very well-positioned with I believe pretty much fully hedged production for this year, but I*'m wondering if at $2 per MCF gas,* you*'re actually starting to see the opportunity to perhaps take away some of the rigs and refocus them in the* Permian where you keep strongly growing the activity. Thank you.". FlanT5 therefore not only marks multiple onset points, but those may also interrupt ongoing sequences.

In brief, T5 is the most reliable model for onset position prediction (offset mean $= 0.93$); FlanT5 is the best at predicting pattern length (offset mean $= -6.2$), as confirmed by similar length distribution in Figure 2f compared to the gold standard of Figure 2a. For further insight, we refer to the overview of Figure 2 (Appendix B).

5.2 Text-to-text generation

The two varieties of BART models were the best performing across metrics. They show similar behavior and the *large* configuration mostly hits some of the misses of the *base* configuration (cf. Table 10). The right starting point is detected in 45% of cases by BART base, increasing to 60% for BART large. The distribution of predicted lengths was the same across both varieties; this means that the *base* configuration is already powerful enough to pick up such a feature to the best that this model family allows given the quantity of training data available. In conclusion, both BART models learned to identify the start of the pattern in the vast majority of

Model	Euclidean distance	SBERT similarity	BERTScore Precision	BERTScore Recall	BERTScore F1
BART (base)	**0.42**	0.07	0.91	0.95	0.93
BART (large)	0.46	**0.08**	**0.92**	**0.96**	**0.94**
T5 (small)	0.46	0.05	0.86	0.93	0.90
T5 (base)	0.59	0.06	0.89	0.94	0.91
T5 (large)	0.54	0.07	0.84	0.90	0.87
GPT-4o	*0.35*	***0.78***	***0.93***	*0.95*	***0.94***

Table 4: Semantic evaluation for text-to-text generation. The best models are shown in bold, second best underlined. The LLM baseline is in italic; additionally, baseline results are in bold if their value is equal or better than the best result.

Model	Γ
BART (base)	**0.56**
BART (large)	0.54
T5 (small)	0.07
T5 (base)	0.26
T5 (large)	—
GPT-4o	***0.69***

Table 5: Annotation agreement evaluation for text-to-text generation. The best models are shown in bold, second best underlined. The LLM baseline is in italic; additionally, baseline results are in bold if their value is equal or better than the best result.

cases; remaining errors, however, greatly diverge from the gold standard and both models tend to considerably overgenerate in the majority of cases (by 77 tokens on average for BART base, 73 for BART large).

T5-large is, conversely, a case of extremely flawed generation: despite all safeguards implemented, none of the retrieved patterns corresponds to a substring of the original text–hence hindering the calculation of the Γ metric in Table 5. For example, compared to the true pattern "Are you suggesting that you could potentially ship to Russia later this year?", the corresponding generation reads: "- And then my follow, as it is in terms of Europe. I just want to clarify that? So this has the potential risk from Russia for approximately 100 million.".

6 Conclusions and future work

This paper introduces a prototypical argumentative pattern that originates in the questions asked during the Q&A sessions of financial dialogues, called the Request Of Confirmation Of Inference (ROCOI). Since argumentation is a pivotal aspect of human communication, the identification and extraction of argumentative patterns is argued to be fundamental in the study of language in interaction. Particularly, given that the identification of argumentative patterns is a challenging yet doable task for trained humans, this study seeks to answer the question of whether language models can perform this task as well.

We adopted two concurrent ML approaches to the extraction of ROCOIs from a wider interrogative unit: sequence labeling and text-to-text generation. The sequence labeling approach, evaluated both at the token- and span-level, shows that FlanT5 is the best-performing model. Qualitative observation of the results, however, marks its outputs as potentially unreliable. T5 is therefore the best-performing model both for accuracy and reliability of the output. The text-to-text generation approach identifies BART-large as the best-performing model across syntactic, semantic, and annotation agreement evaluation measures.

GPT-4o was identified as the state-of-the-art representative for the decoder-only category of language models: appropriate for the task due to its power and despite its limited reasoning abilities–as pattern extraction is not formulated as a reasoning task. The LLM performed poorly in the sequence labeling task in terms of pattern identification, whereas it represents state-of-the-art for extractive generation. The increment in performance, however, does not appear to hold effective positive correlation with model size and its use price. Consequently, we do not deem the LLM baseline as the winning model–especially due to its unreliability across tasks.

In conclusion, this task can be carried out by language models. At the present stage, results suggest that sequence labeling is still the most trustworthy method to approach the task. While results would improve with a larger training dataset, gathering additional samples containing ROCOIs is difficult due to their low absolute frequency (although it represents a relatively frequent argumentative pattern in the ECC context).

Further work may include the insertion of intermediate steps to fine-tune for similar tasks (such as argumentative sequence labeling) before applying them to ROCOI extraction (van der Meer et al., 2022), alongside cross-domain extraction and cross-pattern comparison in extraction performance. Additionally, ROCOI retrieval may enhance current argument mining techniques (D'Agostino, 2025). Type 1 ROCOIs, in fact, always explicitly include the conclusion of a reasoning instance. Even if the rest of the inferential process was omitted from the conversation (i.e., they are enthymemic), the acknowledgment of the ROCOI functions as a placeholder that marks where an inference was drawn in the conversation–thus supporting the retrieval of argumentative instances.

Limitations

Our work has several limitations to consider. While we carefully selected the models for fine-tuning that are open source and accepted baselines among related work in Argument Mining literature, our choice of model architecture remains limited. Further, our relatively limited dataset size affects the generalizability of our results, especially in cases of context shift. Training models with more data or increasing the size of the evaluation set may paint a different image of the relative performance among models. Despite using fixed model checkpoints and consistent dataset splits, we observed that T5's generation outputs exhibit high predictive variability. In addition, we found that FlanT5 has a systematic tendency to overpredict multiple ROCOI spans within individual samples, potentially inflating our metrics.

Consideration must also be given to the inherent limitations in the formulation of the current task. All instances fed to the models did contain at least one ROCOI by design–as the experimental setup assumes the availability of candidate question units, and considers their identification an upstream task (see D'Agostino and Rocci, 2024). However, it is true that the current study neither accounts for guardrails against potential error propagation, nor explicitly handles cases that would entail empty generation (alternatively, fully "O" labeling) as the correct output. This can be addressed with the development of a pipeline that performs pattern identification before its extraction.

Lastly, the span length of the gold standard annotations over which a ROCOI develops is not a settled matter–in fact, IAA is only fair with value $\Gamma = 0.52$. The gold standard against which models are tested in this experimental setup is a pairwise expert curation of such annotations until almost perfect agreement was achieved. Additionally, two ROCOI configurations were defined: the *minimal* and *maximal* extension of the pattern–the latter typically including a phrase or sentence that contains a premise to the conclusion that constitutes the core of a ROCOI (e.g., *minimal ROCOI*: "should we assume that Premier Agent revenue growth should be more muted for the remainder of the year?", *maximal ROCOI*: "But given both of those are likely to remain challenges for at least the remainder of 2022, should we assume that Premier Agent revenue growth should be more muted for the remainder of the year?"). Experiments were conducted on both settings; this study only reports on the minimal setting, as it was the one consistently achieving better results. Further studies will also include refinement of the characterization of the maximal ROCOI extension and a comparison of the retrieval of the two varieties.

Ethical Considerations

Recognizing argumentative content can be biased to the content of the training set. This may result in predictions that are poor in novel contexts or edge cases. Responsible implementations of an extraction system, especially in the financial domain, should always be checked by a human. Our work is a first attempt at creating a system for analyzing argumentative patterns for financial dialogues. Situating our approach in an ecosystem that contains checks and balances will not only ensure responsible use of the predictive model but also may yield valuable insights into the actual use of the model.

Supplementary materials availability statement

The dataset on which these experiments were conducted is freely available on GitHub: https://github.com/dagosgi/ROCOIs/tree/main/LARP2025

Acknowledgments

The work in this paper was in part supported by the Swiss National Science Foundation under the project "Mining argumentative patterns in context. A large scale corpus study of Earnings Conference Calls of listed companies" (grant n. 200857).

References

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6354–6364, Online. Association for Computational Linguistics.

Giulia D'Agostino. 2025. *A framework for the large-scale analysis of argumentative patterns in financial discourse*. PhD thesis, Università della Svizzera italiana, Lugano, Switzerland.

Giulia D'Agostino and Andrea Rocci. 2024. Argumentative patterns in the context of dialogical exchanges in the financial domain. In *Proceedings of the 24th Edition of the Workshop on Computational Models of Natural Argument (CMNA 24)*, Hagen, Germany.

Adrian de Wynter and Tangming Yuan. 2024. "I'd Like to Have an Argument, Please": Argumentative Reasoning in Large Language Models. In *Computational Models of Argument*, pages 73–84. IOS Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.

Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the first workshop on argumentation mining*, pages 39–48.

Deniz Gorur, Antonio Rago, and Francesca Toni. 2024. Can large language models perform relation-based argument mining? *arXiv preprint arXiv:2402.11243*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Katherine Keith and Amanda Stent. 2019. Modeling financial analysts' decision making via the pragmatics and semantics of earnings calls. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 493–503, Florence, Italy. Association for Computational Linguistics.

Klaus Krippendorff. 1970. Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement*, 30(1):61–70.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4691–4698, Florence, Italy. Association for Computational Linguistics.

John Lawrence and Chris Reed. 2019. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Costanza Lucchini, Rocci Andrea, and Elena Battaglia. 2024. Epistemic and Evidential Expressions as Context-Specific Argumentative Indicators in Institutional Dialogues: A Corpus Study of Interactions in the Financial Domain. In *Proceedings of the Tenth Conference of the International Society for the Study of Argumentation*, pages 590–603.

Costanza Lucchini and Giulia D'Agostino. 2023. Good answers, better questions. Building an annotation scheme for financial dialogues. Technical report.

Fabrizio Macagno and Douglas Walton. 2014. Argumentation schemes and topical relations. In Giovanni Gobber and Andrea Rocci, editors, *Language, reason and education*, pages 185–216. Peter Lang, Bern.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The Unified and Holistic Method Gamma (Γ) for Inter-Annotator Agreement Measure and Alignment. *Computational Linguistics*, 41(3):437–479.

Johanna Miecznikowski. 2020. At the juncture between evidentiality and argumentation. *Journal of Argumentation in Context*, 9(1):42–68.

Elena Musi and Andrea Rocci. 2017. Evidently epistential adverbs are argumentative indicators: A corpus-based study. *Argument & Computation*, 8(2):175–192.

Andreas Peldszus and Manfred Stede. 2013. From Argument Diagrams to Argumentation Mining in Texts. *International Journal of Cognitive Informatics and Natural Intelligence*, 7(1):1–31.

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. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Andrea Rocci and Carlo Raimondo. 2018. Dialogical Argumentation in Financial Conference Calls: The Request of Confirmation of Inference (ROCOI). In *Argumentation and Inference: Proceedings of the 2nd European Conference on Argumentation*, pages 699–715. College Publications.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task Chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Michiel van der Meer, Enrico Liscio, Catholijn Jonker, Aske Plaat, Piek Vossen, and Pradeep Murukannaiah. 2024. A hybrid intelligence method for argument mining. *Journal of Artificial Intelligence Research*, 80:1187–1222.

Michiel van der Meer, Myrthe Reuver, Urja Khurana, Lea Krause, and Selene Báez Santamaría. 2022. Will it blend? mixing training paradigms & prompting for argument quality prediction. In *Proceedings of the 9th Workshop on Argument Mining*, pages 95–103.

Frans van Eemeren and Rob Grootendorst. 2004. *A Systematic Theory of Argumentation: The Pragma-dialectical Approach*. Cambridge University Press.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Experimental details

A.1 Training parameters

We present additional details regarding the usage of pretrained models for the two formulations of the ROCOI extraction. We present an overview of the initial model checkpoints and their parameter counts in Table 6. The hyperparameters to train the models on the sequence labeling task are given in Table 7, and the ones for text-to-text generation are given in Table 8. Training a single model generally takes up to one hour at most on modern hardware (one RTX3090 or A100 GPU).

Model	Checkpoint	Size
BERT (base)	`google-bert/bert-base-uncased`	109M
SpanBERT	`SpanBERT/spanbert-base-cased`	108M
TinyBERT	`huawei-noah/TinyBERT_General_4L_312D`	14M
T5 (base)	`google-t5/t5-base`	110M
Flan-T5 (base)	`google-t5/flan-t5-base`	110M
BART (base)	`facebook/bart-base`	139M
BART (large)	`facebook/bart-large`	406M
T5 (small)	`google-t5/t5-small`	61M
T5 (base)	`google-t5/t5-base`	223M
T5 (large)	`google-t5/t5-large`	738M

Table 6: Description of each model and the specific checkpoint we used.

Sequence labeling For the sequence labeling models, we train on the training set (75% of total available samples) while observing metrics on a validation set (10% of samples). We pick the model iteration with the highest token-level F_1 score and evaluate that model on the test set (15% of samples) to obtain the results reported in Tables 1 and 2. We use the same split for each experiment.

Model	Parameter	Value
BERT (base)	learning rate	2e-05
SpanBERT	learning rate	2e-05
TinyBERT	learning rate	2e-05
T5 (base)	learning rate	4e-04
Flan-T5 (base)	learning rate	4e-04
all	batch size	16
all	max sequence length	256
all	max epochs	100

Table 7: Hyperparameters for the sequence labeling approaches.

Text-to-text sequence generation For the text-to-text generation models, we train on the training set (75% of total available samples) while observing metrics on a validation set (10% of samples). We optimized hyperparameters and picked the best model iteration with the lowest loss value, and evaluated that model on the test set (15% of samples) to obtain the results reported in Tables 3, 4, and 5. We use the same split for each experiment.

B Error analysis

We present additional details upon which we based our qualitative observations of Section 5. Particularly, we display the the raw numerical data for each test instance, which in the body of the paper was instead merged in the form of percentage over the total. Table 9 refers to the sequence labeling task and reports

Model	Parameter	Value
all	learning rate	6e-06
BART *all*	batch size	4
T5 *(all)*	batch size	6
all	max sequence length	256
all	max epochs	100

Table 8: Hyperparameters for the text-to-text approaches.

begin- and length- offsets of the predicted patterns with respect to the gold standard, alongside the number of ill-formed sequences in the tag sequence. Table 10 presents begin- and length- offset numbers only, from the text-to-text generation task. Finally, Figure 2 displays the differences in predicted ROCOI lengths across models for the sequence labeling approach, compared to gold standard.

begin offset	length offset	ill-formed sequences
0	-2	0
n.a.	n.a.	0
0	0	0
0	0	0
n.a.	n.a.	1
0	-5	0
0	-9	0
0	-33	0
0	-2	0
0	0	0
-41	-16	0
n.a.	n.a.	0
0	-34	0
n.a.	n.a.	0
0	3	0
1	-41	0
0	3	0
73	-5	0
-18	-6	0
0	-15	0

(a) T5 (base)

begin offset	length offset	ill-formed sequences
0	-1	4
0	0	4
0	0	3
0	0	1
0	10	2
0	0	3
0	4	2
0	-33	3
0	0	2
0	0	3
n.a.	n.a.	2
0	-1	2
0	-34	3
n.a.	n.a.	2
0	5	3
1	-14	3
12	-5	2
69	-1	2
-18	-6	2
0	-26	4

(b) FlanT5 (base)

Table 9: Qualitative error analysis: sequence labeling approach. Reported the two best performing models. For each sub-table, the first two columns indicate offsets (predicted-true) and the third one indicates the absolute number of instances. The best value is zero for all features.

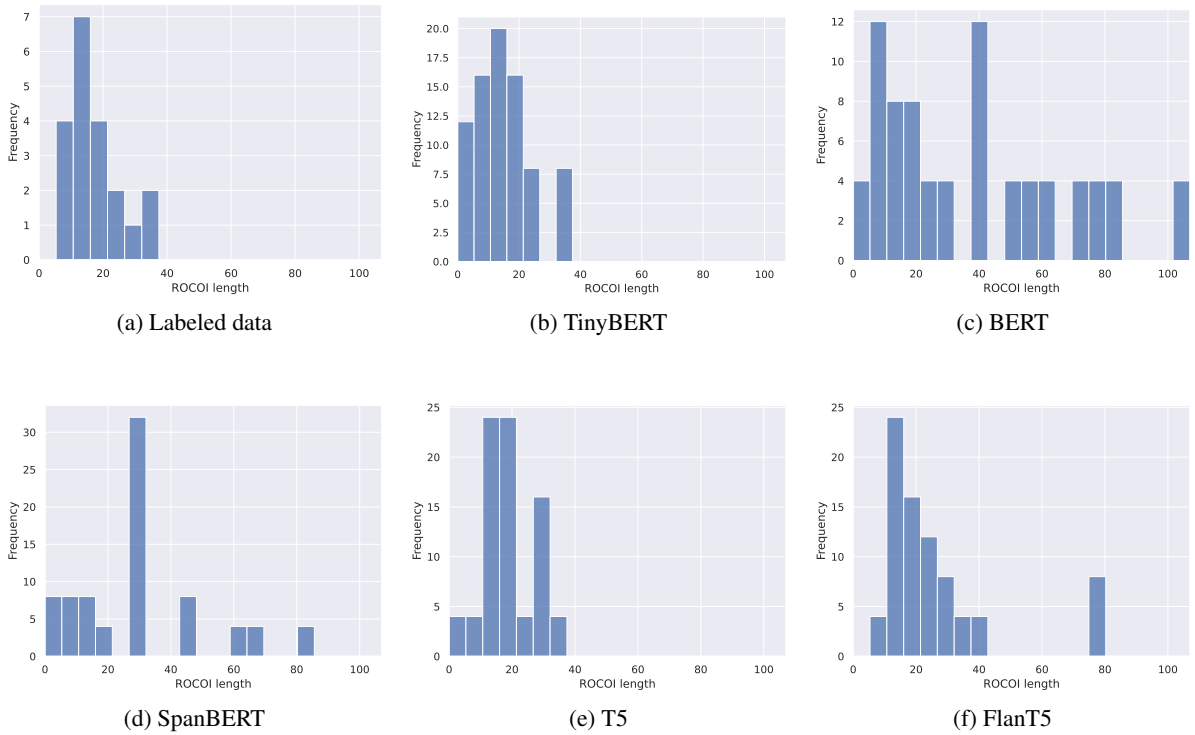
begin offset	length offset
-218	187
0	0
n.a.	n.a.
0	0
0	78
158	5
0	71
-47	47
0	0
-160	77
-179	34
-196	196
0	0
n.a.	n.a.
0	33
n.a.	n.a.
0	70
-4	87
-74	74
0	38

(a) BART (base)

begin offset	length offset
0	182
0	228
n.a.	n.a.
0	0
0	78
158	5
0	22
-47	47
0	0
0	35
-179	34
0	0
0	0
-186	85
0	33
n.a.	n.a.
0	70
-4	87
n.a.	n.a.
0	38

(b) BART (large)

Table 10: Qualitative error analysis: text-to-text sequence generation approach. Reported the two best performing models. For each sub-table, the two columns indicate offsets (predicted-true). The best value is zero for all features.



(a) Labeled data (b) TinyBERT (c) BERT

(d) SpanBERT (e) T5 (f) FlanT5

Figure 2: Error analysis: sequence labeling approach. True (upper left) and predicted (others) ROCOI lengths.

Author Index

Ali, Manar, 38

Berman, Alexander, 28
Buschmeier, Hendrik, 38

D'Agostino, Giulia, 51

Junker, Simeon, 38

Larsson, Staffan, 28

Maraev, Vladislav, 28
Mathai, Ved, 11
Mompelat, Ludovic, 1

Pierrehumbert, Janet B., 11

Reed, Chris, 51

Sarzotti, Marika, 38

Van Der Meer, Michiel, 51

Zarrieß, Sina, 38