When Only Time Will Tell: Interpreting How Transformers Process Local Ambiguities Through the Lens of Restart-Incrementality

Anonymous ACL submission

Abstract

Incremental models that process sentences one token at a time will sometimes encounter points where more than one interpretation is possible. Causal models are forced to output one interpretation and continue, whereas models that can revise may edit their previous output as the ambiguity is resolved. In this work, we look at how restart-incremental Transformers build and update internal states, in an effort to shed light on what processes cause revisions not viable in autoregressive models. We propose an interpretable way to analyse the incremental states, showing that their sequential structure encodes information on the garden path effect and its resolution. Our method brings insights on various bidirectional encoders for contextualised meaning representation and dependency parsing, contributing to show their advantage over causal models when it comes to revisions.[1]

1 Introduction

This is the honey... even if we stopped mid-sentence here, you would likely have created a partial interpretation of this prefix considering *the honey* as (the beginning of) a noun phrase. It could have many continuations, *e.g. that skunks like*, or *produced by stingless bees*. But what if the next token is another noun, as *bee*? A semantic parser would have to revise its previous hypothesis to accommodate the fact that *honey* has become a modifier of *bee*.

Bidirectional NLP models (*i.e.* those that encode linguistic input using both its left and right context) have transformed the field of computational linguistics. But that has come at the cost of cognitive plausibility in various aspects, in particular establishing a disregard for language's temporal structure. While humans process language one increment at a time (Marslen-Wilson, 1973; Altmann and Steedman, 1988; Levelt, 1993), BiLSTMs (Graves and Schmidhuber, 2005) and Transformers (Vaswani et al., 2017) have mechanisms that rely on access to the complete input sequence, making them unsuitable as off-the-shelf components for incremental applications. Although their unidirectional counterparts, LSTMs and autoregressive (*aka* causal) Transformers, are equipped with the possibility of incremental decoding (making predictions relying only on left context), their token representations are static and thus not updated as incoming tokens arrive (Eisape et al., 2022), because the underlying encoding is unidirectional. This places them at the disadvantage of not being able to revise, which is a desired property to recover from mistakes or local ambiguities (Schlangen and Skantze, 2011; Madureira et al., 2023), as shown in Figure 1.

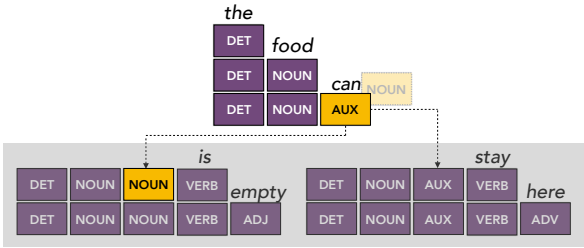


Figure 1: A prefix with multiple valid continuations. A causal decoder is forced to output only one POS-tag for the token *can* at this point and cannot change it anymore, even if its internal state encodes the local ambiguity. In contrast, a restart-incremental model can perform revisions and would thus be able to recover if the selected label turned out to be incorrect (as in left).

The restart incremental (RI) paradigm[2] (Schlangen and Skantze, 2011) circumvents this issue by adding an interface upon any model, making it work incrementally by processing prefixes from scratch whenever a new input increment arrives (Beuck et al., 2011). Even though it is computationally costly, RI has been effectively studied in simultaneous MT (Arivazhagan et al., 2020; Sen et al., 2023), dialogue systems (Khouzaimi et al.,

[1] Code: https://anonymous.4open.science/r/restart-inc-56A0/

[2] Or qualitative incrementality (Kilger and Finkler, 1995).

2014), disfluency detection (Chen et al., 2022) and NLU pipelines (Rafla and Kennington, 2019). Besides, it has the advantage of incorporating recent increments into past predictions, revising hypotheses when deemed necessary.

A recent line of work has investigated restart-incrementality for sequence labelling by profiling bidirectional encoders (Madureira and Schlangen, 2020; Kahardipraja et al., 2021) and modelling and evaluating revision policies (Kahardipraja et al., 2023; Kaushal et al., 2023; Madureira et al., 2023). Still, evaluations so far have been performed in a black box fashion: Only the relations between input tokens and output labels have been considered. The output labels as top positions of the softmax operation are only one product of many intermediate computations that have not been examined.

Therefore, we still do not know why RI models revise when they revise. What happens internally in the model's mechanisms to encode the need to edit previous outputs? How can (static) causal representations be affected under RI bidirectionality? Can we predict whether the model will recover from a wrong interpretation? To answer such questions, we need a shift to glass box interpretability methods that can shed light on the dynamics of updates in the internal states which lead to output revisions. Linguistically motivated analyses are also required to examine the behaviour of bidirectional models with respect to specific phenomena known to cause reinterpretations, like garden path constructions.

Thus, our present contributions to make RI models more transparent and explainable are (a) a formalisation of RI sequential processors as transition systems that create structured step-by-step constructions not present in causal models; (b) a proposal of interpretability methods for the internal mechanisms of RI models; and (c) an analysis of the strategies employed by various models on stimuli containing local ambiguities, for which a well-defined motivation for reanalysis is known.

2 Related Work

Monotonic decoding is advantageous to avoid output instability but comes with the downside of not recovering if there is a genuine reason to revise. Some attempts to overcome this issue are adapting back-propagation to update the internal representation of the output (Qin et al., 2020) or gradient-based methods to update the cached internal representations (Yoshida and Gimpel, 2021) without changing the model's parameters. Although this alleviates the strict monotonicity, it requires adapting the model's implementation. RI, on the other hand, is available to anyone in possession of any model, as it only requires re-running it as is each time. Via recomputations, bidirectional models innately incorporate new increments into its states and revise if needed (Kahardipraja et al., 2023).

Several methods have been proposed to analyse how neural networks encode linguistic information (Belinkov and Glass, 2019), with Transformers front and centre, *e.g.* in how information flows in its self-attention (Abnar and Zuidema, 2020) and how its predictions are refined or affected by previous tokens layer after layer (Ferrando et al., 2023; Belrose et al., 2023; Din et al., 2023). Voluminous BERTology works exist (Clark et al., 2019; Rogers et al., 2020), in particular investigating its attention mechanisms, what it means for an embedding to be contextualised instead of static and what layers encode what type of linguistic structure. Another angle is to look at how linguistic information is dynamically learnt throughout the training regime of LMs (Saphra and Lopez, 2019; Chen et al., 2023).

The realm of incremental processing has also been active. Ulmer et al. (2019) propose methods to interpret how RNNs incrementally encode, integrate and retain information. The incremental aptitude of neural networks to encode syntactic information has also been examined with the aid of psycholinguistic experimental methodologies (Futrell et al., 2019). An evident test bed are stimuli containing local ambiguities to evaluate if they exhibit garden path effects. In psycholinguistics, a traditional debate holds between the early-commitment with eventual reanalysis *vs.* beam search approaches, where multiple hypotheses are kept in parallel, known as two and one-stage accounts, respectively (Van Schijndel and Linzen, 2021). In this sense, the RI paradigm can be viewed as a two-stage account performing multiple forward-reanalyses (Frazier and Rayner, 1982).

Recent findings point to models encoding multiple hypotheses in face of local ambiguities. Aina and Linzen (2021) probe to what extent autoregressive LMs encode multiple syntactic analyses by assessing the probability assigned to each interpretation as they generate continuations of a prefix, finding that multiple interpretations are followed in parallel. Methods relying on surprisal theory show that the magnitude of the garden path effect in autoregressive and RNN LMs underestimate human

behaviour (Van Schijndel and Linzen, 2021; Arehalli et al., 2022). Other works study BERT-based models using differences in surprisal and attention between unambiguous and locally ambiguous sentences (Lee et al., 2022; Lee and Shin, 2023).

Other glass box analyses have been put forward. Irwin et al. (2023) use probing tasks with garden paths to study how the BERT family assigns semantic roles in QA tasks. Jurayj et al. (2022) define vector similarity methods as an improvement over surprisal to shed light on how GPT-2 traverses garden paths, finding periods of ambiguity in the hidden states that do not always surface. Lindborg and Rabovsky (2021) study how meaning is built word by word by GPT-2 looking at the connection between the size of the updates in its output activations and the N400 psychometric in humans.

Through structural probes, Hewitt and Manning (2019) show that it is possible to recover syntax trees from ELMo and BERT embeddings using linear transformations. Eisape et al. (2022) extend the probe to incremental settings and conclude that the internal representations of autoregressive LMs encode syntactic uncertainty that can be explored by future tokens, which would be a reason why such models perform well even without access to future words. It is also possible that monotonic models work well due to speculation about the future. Pal et al. (2023) use GPT's internal states to predict future tokens, finding that some layers partially anticipate subsequent tokens. But this speculation may lead to wrong paths and is not always desirable. Kitaev et al. (2022), for instance, explore non-speculative incrementality to induce syntactic representations free of speculation.

As we see, in previous studies, analyses happened either at autoregressive mode or with bidirectional access to the whole sequence. Our focus is to tailor interpretability methods for inspecting the emergent properties of bidirectional models under *restart incrementality*, beyond output labels.

3 Formalisation

A restart-incremental model is constructed upon an underlying (non-incremental) model, making it perform a sequence of re-computations as the input is processed increment by increment (Schlangen and Skantze, 2011; Beuck et al., 2011). We propose a general formalisation of this procedure, detail its structures for sequential processing and discuss their interpretation for bidirectional models.

Restart-Incrementality Let $M : w \mapsto o$ be a (non-incremental) model that maps an input w to an output o by computing internal states s. Generally speaking, restart-incrementality is an interface I around M, such that $I(M)$ can be fed individual tokens w_i, which—until the interface is reset—are taken as continuations of the sequence of tokens fed so far. Hence, at time step t, $I(M)$ will be provided with the token w_t. Internally, the interface assembles the *prefix* $w_1, \ldots, w_t$ (denoted as w^t below), which it feeds to M, to compute the corresponding sequences $s^t = (s^t_1, \ldots, s^t_t)$ and $o^t = (o^t_1, \ldots, o^t_t)$.[3]

This way, from the perspective of M, it is always a sequence of tokens that is being processed from scratch, independently from any prior calls of M, while from the perspective of $I(M)$, with each call a single token is added. While for the further downstream processing, only the output revisions (that is, the elements where o^{t+1} differs from o^t) may be of interest,[4] for the purpose of our analysis the sequences $s^1, s^2, \ldots, s^n$ corresponding to each prefix are kept in memory by $I(M)$. $I(M)$ can hence be regarded as a transition system where states are updated based on the action of recomputing with the integration of a new input increment.

Triangular Structures To represent the memory built by $I(M)$, we can extend the 2D chart of outputs in Madureira et al. (2023) with a third dimension, filling it with state sequences as follows:

$$
\begin{array}{ccccc|ccccc}
 & & & & w_1 & s^1_1 & & & & \\
 & & & w_1 & w_2 & s^2_1 & s^2_2 & & & \\
 & & w_1 & w_2 & w_3 & s^3_1 & s^3_2 & s^3_3 & & \\
 & \iddots & \iddots & \iddots & \vdots & \vdots & \vdots & \vdots & \ddots & \\
w_1 & w_2 & w_3 & \ldots & w_n & s^n_1 & s^n_2 & s^n_3 & \cdots & s^n_n
\end{array}
$$

The right portion represents how the states evolve from time step to time step for an input sequence of n tokens. The last state sequence coincides with the sequence M creates when the full input is available. Figure 2 illustrates three types of resulting structures. Although they are actually multidimensional arrays, with some abuse of nomenclature, we will reference them by the triangular prisms they resemble. In the right triangle

[3] By growing incrementally, each prefix changes the context of each prior input token, which is why the elements of state and output sequences need to be indexed with the timestep in superscript; e.g., s^4_1 is the internal representation of token 1, at the time when the first 4 tokens were available.

[4] As in the IU model of Schlangen and Skantze (2011), implemented in InproTK (Kennington et al., 2014).

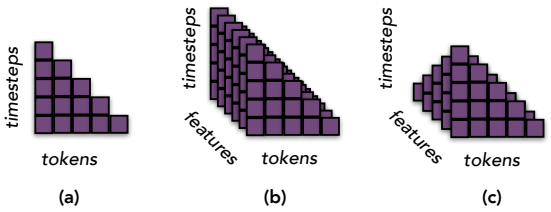


Figure 2: Triangular structures representing states built step by step in restart-incremental sequential processing.

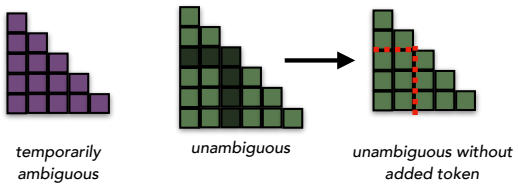


Figure 3: We realign tokens to compare a locally ambiguous sentence with its unambiguous counterpart.

in (a), each token is assigned one value per time step. In the right triangular prism (b), each time step produces a vector with a fixed number of dimensions for each token (for instance, a probability distribution over labels or an embedding). In the truncated triangular prism (c), at each time step, a vector the same size of the current prefix is built for each token (for example, attention scores or dependency arcs over the input).

These structures allow us to inspect emergent properties by comparing time steps. We can specifically look at the dynamics of the s_k^j for a fixed k, *i.e.* the states corresponding to the same input token, but over the re-computations at each time step. Note that, when models are bidirectional, this is a much richer process than what happens in autoregressive or monotonic decoders, where elements in the main diagonal are computed once and kept fixed for all subsequent steps. Here, all s can keep changing as right context gets integrated.[5]

4 Method

We now introduce our method and motivate its interpretations. Once we have extracted states as a RI model processes a sentence, we can analyse the dynamics of this process, *i.e.* how do these vectors evolve over time. The 3D structures (b) and (c) from Figure 2 can be converted into 2D as in (a) if we apply a metric over the features dimension, summarising it into one value (*e.g.* Shannon's entropy or divergence over distributions and similarity or distance metrics for embeddings). With such scores, we can then examine whether the behaviour of their variation aligns with output edits.

If we look at columns of the triangular structures, we can analyse a sequence of states assigned to one token step by step. If we consider the rows, we see the effect that the recently added token had on the current prefix. This analysis can in principle be applied to any sentence. However, to shed light on how restart-incremental models edit the output and how they engage in reinterpretation, it is useful to analyse what happens when the linguistic input is known to contain local ambiguities. That way, we have a genuine motivation to expect revisions and can study how states change at key positions.

Let v_{jk}^i be the value assigned for feature k of token j at time step i in a sequence with n tokens. We can compare the differences between v_{jk}^i and four relevant positions with clear interpretations:

- the value v_{jk}^j at the main diagonal, *i.e.* the initial interpretation of token j when it was first observed without any right context
- the value $v_{jk}^{(i-1)}$ at the previous time step, *i.e.* the change in the interpretation of token j caused by the most recently added token w_i
- the value v_{jk}^n at the last time step, *i.e.* the final (gold standard) interpretation of token j when it is observed with the whole context
- the value $v_{jk}^{(i+d)}$, where $i + d$ is the position when the disambiguating token occurs

A classic approach in studies using surprisal or reading time as a measure of processing difficulty is to consider the difference of the measure in the disambiguating region of an unambiguous sentence and a locally ambiguous sentence as the garden path effect (Van Schijndel and Linzen, 2021; Arehalli et al., 2022; Huang et al., 2023). Drawing a parallel with that approach, we compute triangular structures for the stimulus and its baseline. After removing the extra tokens, we have two structures with the same dimensions, as shown in Figure 3. To give an example, such a pairing would be *"The professor noticed the grant gained more attention"* and *"The professor noticed (that) the grant gained more attention"*, with the token in parentheses removed **after computing the states** to make the structures directly comparable.

[5]The main diagonal contains another form of causality: Although its elements also see no right context, they are computed considering bidirectional representations of the left tokens available. In causal models, all states are unidirectional.

5 Analysis

Overall, we are interested in how future tokens affect states of past tokens and how such changes surface as output revisions. Our main hypothesis is that RI models are led down the garden path when they first encounter a local ambiguity without further right context but, as the disambiguating region is integrated, states are updated to absorb the new interpretation. We provide insights into updates that may be allowing them to recover.

Scope We apply our method to two RI scenarios. Firstly, we look at the construction of meaning representations in bidirectional LMs (§5.2). We assume that the representations of a token encode its meaning in the available context, tracking how it evolves step by step and across all the layers. If there is a shift in the meaning of a token, we expect to observe variation over a control reference in the corresponding states. The second scenario is dependency parsing as sequence labelling with arcs and relations (§5.3) (Spoustová and Spousta, 2010; Strzyz et al., 2019). In this case, we have the output labels as an external signal of the concrete decisions made by the model. We investigate whether changes in divergence of the attention scores and of the distribution over labels or arcs align with output edits and with the resolution of the ambiguity.

Material We study three kinds of local ambiguities in English: 24 instances of Direct Object/Sentential Complement (NP/S) and of Main Verb/Reduced Relative ambiguity (MVRR) garden paths from Huang et al. (2023) and 281 instances of noun-noun compounds (NNC) from Garcia et al. (2021) with a fixed context. Examples of each type are shown in Figure 4. NNC has a very localised need for revision, where the immediate next token changes the interpretation of the preceding noun. In NP/S and MVRR, the disambiguating token appears after a NP, with a more broad syntactic and semantic shift of all the prefix. Please see the original publications for detailed motivations.

NNC	(s)	*This is the **public** service ,*
	(b)	*This is the public ,*
NP/S	(s)	*The professor **noticed the grant** gained more attention…*
	(b)	*The professor noticed that the grant gained more attention…*
MVRR	(s)	*The dancer **assigned** the ballet achieved incredible success…*
	(b)	*The dancer who was assigned the ballet achieved incredible…*

Figure 4: Types of stimuli (s) and their reference baselines (b). The words in lilac/bold are locally ambiguous until the underlined/yellow token is observed.

5.1 General Effects of Right Context

We start by showing, in Table 1, that even in the baseline stimuli (where no major revisions are expected), a token's state keeps being updated as more right tokens are integrated. For meaning, we compute the cosine distance between a state and its previous version at the last layer of BERT, to isolate how the latest token w_{t+i} affects s_t^{t+i-1}. For dependency parsing, we measure the variation of entropy of the arc distribution also with respect to its previous state for the biaffine parser (Dozat and Manning, 2017) with RoBERTa. On average, there is a considerable effect when the immediately next token is added, which decreases gradually as more right context is observed in almost all cases.

	$t{+}1$	$t{+}2$	$t{+}3$	$t{+}4$	$t{+}5$	$t{+}6$	$t{+}7$
Meaning							
MVRR	0.38	0.09	0.05	0.04	0.04	0.03	0.03
NPS	0.39	0.10	0.07	0.05	0.03	0.03	0.02
NNC	0.34	0.12	0.15	0.13	-	-	-
DP							
MVRR	0.11	0.02	0.01	0.02	0.01	0.00	0.00
NPS	0.19	0.06	0.05	0.01	0.00	0.00	0.00
NNC	0.14	0.21	0.29	0.08	-	-	-

Table 1: Average effect of token w_{t+i} on s_t over all baseline stimuli.

5.2 Incremental Construction of Meaning

For this part, we extract the hidden states for all layers of pretrained bidirectional transformer LMs. We show results for BERT (Devlin et al., 2019) here and RoBERTa (Liu et al., 2019b) in the Appendix. We also study static embeddings in causal models, namely GPT-2 (Radford et al., 2019) here and OPT (Zhang et al., 2022) in the Appendix. We measure how much states are updated, by computing their cosine distance to a reference time step.

NNC In Figure 5, we zoom in at what happens to the prefix when the second noun is added (the fifth time step) by computing the cosine distance between states s_i^4 and s_i^5, for $i = 1, \ldots, 4$. We subtract from it a baseline case where a comma is observed instead. This controls for keeping the meaning of the first noun as a NP head versus it becoming a modifier of the second noun. The results show that the second noun affects the meaning of all previous tokens, more than the baseline, and the effect is even larger for the first noun, in all layers but especially in middle ones. In the last layer, the mean cosine distance between the initial state of

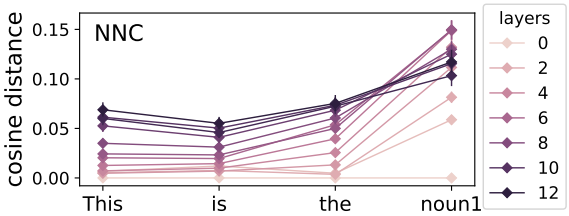


Figure 5: BERT's mean effect of the second noun on the tokens in the prefix. Absolute difference over baseline.

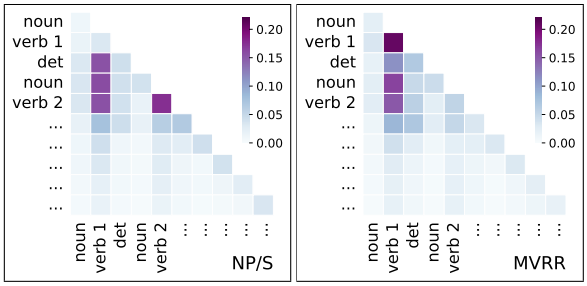


Figure 6: Average absolute difference between stimulus and baseline in distance of each representation to its final state at layer 9 of BERT.

the first noun and its updated version when the second noun is observed is almost 0.41, considerably above the corresponding 0.34 in Table 1.

NP/S In these stimuli, humans can first interpret the NP as direct object, but only until disambiguating region is observed, when it becomes a sentential complement. The final meaning of the stimulus is the same as the baseline's.Thus here we measure how distant each s_i^j is from its target interpretation s_i^N at the final time step N. Then we compute the absolute difference between the distances for the stimulus and the baseline. In Figure 6 (left), we show the effect around the disambiguating token for the layer where it is most prominent. The representation of the first verb diverges from its final meaning as the NP is processed. After the disambiguating region is integrated, this difference almost disappears. This suggests that the model first builds an initial interpretation for the stimulus but, one token after the second verb, it turns to its final meaning, as in the unambiguous case. BERT seems to load the semantics of the argument on the first verb, so the noun does not encode so much what its role is in the two variations of the sentence. We also observe this in almost all layers, especially the middle to upper ones (see Appendix).

MVRR For stimuli where the first verb becomes a reduced relative, we perform the same type of analysis as in NP/S and observe similar results. In Figure 6 (right), a similar pattern occurs, but the divergence starts earlier, at the initial step where the first verb is observed. Again, this is evidence that the initial representation of the verb did not fully encode what would be the final reduced relative form and is revised in the face of upcoming tokens.

The mean variation on the first verb once the second disambiguating token is observed is 0.15 and 0.14 for NP/S and MVRR, well above the average variation at $t + 4$ in Table 1. The conclusion so far is that future tokens do affect the meaning representation of previous tokens and more considerably so when the model has a linguistic motivation to revise its states.

Causal Models We now look at what happens with static embeddings by directly computing the distance from the states in the ambiguous stimulus to the corresponding states in the baseline, which disambiguates the meaning in advance. We subtract from that the distance of a similar pair of sentences with an unambiguous first verb (*given* for MVRR and *said* for NP/S, see Appendix for the detailed formulation) to account for the effect of one sentence having more tokens than the other. The remaining absolute difference is shown in Figure 7. The intermediate layers encode a difference, which should be due to non-commitment in the ambiguous stimulus to what will turn out to be the "true" analysis. That can be interpreted as how the model would revise, if it could. In OPT (Appendix), the difference remains and can affect its predictions. For GPT-2's last layer, however, the distance is close to 0 in both pairs, so that the actual states used for downstream decisions are practically the same in the stimulus and the baseline, despite their potentially different unfolding meanings.

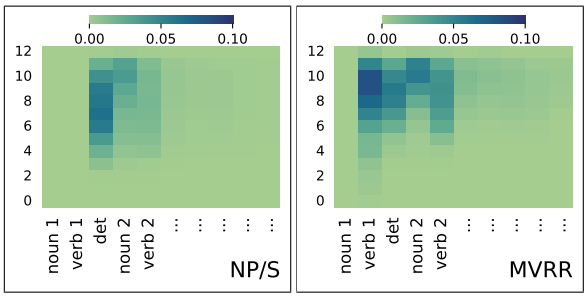


Figure 7: Distance between the causal embeddings of the stimulus and the baseline, after taking the absolute difference of the expected variation by a counterpart pair with an unambiguous verb, for all layers of GPT-2.

5.3 Incremental Dependency Parsing

Aside from being a fundamental aspect of human language processing, incremental parsing is also useful in applications such as simultaneous translation (Ryu et al., 2006) and disfluency detection (Honnibal and Johnson, 2014). We investigate two dependency parsers in a RI setting: 1) the biaffine parser (Dozat and Manning, 2017) trained on top of RoBERTa and fine-tuned on PTB (Marcus et al., 1993) and 2) the DiaParser (Attardi et al., 2021), which uses ELECTRA (Clark et al., 2020) and is fine-tuned on PTB and EWT (Silveira et al., 2014). Here we only show findings for the biaffine parser as we find similar results to occur in DiaParser. See the Appendix for a full comparison.

Our focus is on the self-attention mechanism of the parsers, from which the dependency arcs and labels are directly derived. Both parsers handle parsing and labelling decisions sequentially, selecting the labels for each arc only after ensuring the well-formedness of the tree via the MST algorithm (Chu and Liu, 1965; Edmonds, 1967). As such, an independent analysis of the dependency labels is not possible, as they depend on the predicted arcs, which in turn may also change from time to time as more tokens are observed. Hence we analyse both dependency arcs and labels in a joint fashion. We measure how attention distributions evolve across time steps, by computing the Jensen-Shannon divergence (JSD) with respect to the last, previous, or first (the main diagonal) time step as a reference. We also consider how the arc may change due to future tokens. See the Appendix for full details.

NNC We isolate the effect of the local ambiguity by computing the absolute difference of JSD between the stimulus and the baseline in a similar manner to (§5.2). Using the first time step as a reference, we glean how far the label distribution of the first noun shifts from its origin when it acts as a modifier for the second noun as opposed to encountering a comma, where it retains its interpretation as the NP head. We also compute the difference with respect to the previous step to investigate how the distribution shift causes the dependency structure to change through time. Both are shown in Figure 8. On the left, we see that the second noun alters the original distribution of the first noun more than the baseline, as the head-dependent relation of the first noun changes. To the right, we find that the second noun affects not only the first noun, but also the distributions of all previous tokens as it replaces the first noun as the argument of *is*. This is not present on the left figure.

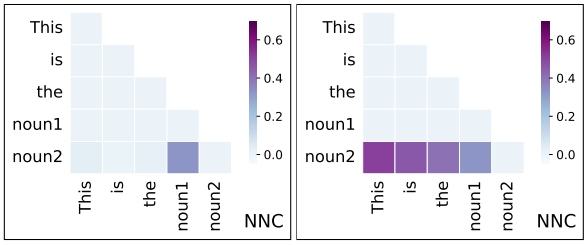


Figure 8: Average absolute difference of JSD between stimulus and baseline on NNC. Left: first step reference. Right: previous step reference.

NP/S As the stimulus in the NP/S ambiguity has the same final interpretation as their unambiguous counterpart, we take the final step as a reference and compute the absolute difference of JSD between them, factoring out the complementiser *that*. We find that the label distribution of the sentential complement diverges at the beginning between the stimuli and the baseline when compared to the final distribution (Figure 9, left). However, there is almost no difference between them after entering the disambiguating region, similar to (§5.2).

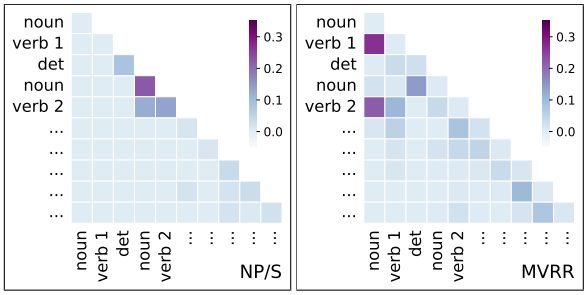


Figure 9: Average absolute difference of JSD between stimulus and baseline on NP/S (left, final step reference) & MVRR (right, previous step reference).

MVRR For MVRR, we are interested in how the dynamics of the parse evolve throughout the time axis. We subtract the JSD of the stimulus calculated wrt. the previous step from the baseline chart. In Figure 9 (right), we notice peaks in the difference for the first noun when the first and second verbs are encountered. The first one occurs as the first verb can initially be parsed either as the main verb or the verb in the reduced relative clause, making the first noun be interpreted differently. However, this is reconciled in the second peak when the disambiguating region is incorporated to the stimulus' interpretation. We observe an

effect similar to NP/S afterwards, as the ambiguity is resolved and the JSD difference vanishes.

Alignment with Edits We examine whether changes in label distributions due to future tokens also coincide with changes in arcs or labels. For instance, this happens in NNC when the first noun is revised from the argument of the verb to be the modifier of the second noun. To do this, we use the JSD chart with the previous step as a reference and assume that changes or edits happens if its value is bigger than a threshold $\tau = 0.45\ln(2)$, where $\ln(2)$ is the divergence upper bound. Then we compute the Matthews correlation coefficient (MCC) (Matthews, 1975) between the predictions and the edits (Table 2). In general, we find that distribution shifts are positively correlated to edits, with a stronger magnitude for dependency arcs compared to labels. This suggests that shifts in distributions encode information that surface as revisions.

	NNC		NP/S		MVRR	
Arc	S	B	S	B	S	B
Biaffine	0.99	0.96	0.95	0.92	0.86	0.93
DiaParser (EWT)	0.97	1.00	0.95	0.92	0.80	0.94
DiaParser (PTB)	0.81	0.78	0.91	0.92	0.79	0.96
Label	S	B	S	B	S	B
Biaffine	0.55	0.77	0.86	0.77	0.71	0.84
DiaParser (EWT)	0.37	0.53	0.78	0.70	0.61	0.75
DiaParser (PTB)	0.23	0.52	0.78	0.72	0.64	0.86

Table 2: MCC between distribution shifts and edits for dependency arcs and labels. S: stimulus & B: baseline.

6 Discussion & Conclusion

Our method has shown that a RI interface applied to bidirectional encoders yields models that build sequences of states with a rich dynamics of updates of past representations. This grants these models with *revisability*, a property that is desirable in incremental systems (Schlangen and Skantze, 2011) but not present in unidirectional models. While causal models must create a representation based only on left context and stick to their first commitment, RI bidirectional models can profit from incorporating right context and revisit its previous decisions. Our analysis has empirically revealed that the RI models we studied seem to run into the downsides of parsing, *i.e.* they are led down the garden path, but their initial internal representations are updated once the disambiguating region is processed, more than in the baselines. In other words, it seems that they make early commitments but then revise accordingly, as two-stage approaches to language processing.

In the analysis of contextualised embeddings, the effect is less pronounced on lower layers, but more prominent from middle to upper layers. This relates to the findings of Tenney et al. (2019) who suggest that upper layers (that are more semantic) can be used to disambiguate decisions in lower layers (that are more syntactic). This is also in line with works showing that middle to upper layers are most informative for some tasks (Liu et al., 2019a).

Treating the triangular structures as internal states of a transition system constructed from graph-based dependency parsers allows us to uncover how relations between tokens change as an effect of future tokens. By using the divergence computed with various reference time steps as a measure, we show that attention distributions may evolve differently throughout the course of processing and how they can become similar again through the disambiguation process. The shift in distributions also indicates edits in dependency arcs and labels, and more importantly, enlightens us more about why RI models revise when they do.

While decoder-only LLMs have grown in popularity the past few years, we still see room for improvement, particularly regarding how their static representations can be updated (like RI models) as more tokens are processed. This can benefit tasks that require non-monotonic reasoning (Kraus et al., 1990), regardless of whether it involves NLG or NLU. Moreover, natural language is inherently ambiguous, although its spectrum of ambiguity depends on its use (Schlangen, 2023) which needs this processing mode.

One of Transformers' properties that does not align with human language processing, is how words are processed in parallel, even in autoregressive settings. Ideally, models of human language should also face similar input and linguistic challenges as humans do (Blank, 2023). We show that when facing such challenge in the form of garden-path sentences, Transformers with a RI interface still exhibit the ability to disambiguate using forward-reanalysis (Frazier and Rayner, 1982).

Future research can explore how the restart activity can serve as an indicator of updates that should not be immediately integrated into the output, or model a controller that is able to detect variations that lead to revisions and decide whether to delay outputs until a level of certainty has been reached.

Limitations

We have only considered short-range temporary ambiguities, *i.e.* those that are resolved up to 3 of 4 tokens into the future. It would be interesting to study whether meaningful revisions also occur when the distance between the temporarily ambiguous token and the disambiguating region is longer.

We observed that the standard deviation of the means we report can be large at some cases. That means that, for some sentences, the effect is less present and in others where it is more extreme. This may relate to what is the most likely early commitment given the lexical information of the ambiguous tokens. Further investigation could try to consider that. Larger samples would also be necessary to have better mean and std. estimators. Related to that, we did not take into account the frequency of the vocabulary choices in these datasets.

The garden path effects we found may not be so salient for all types of local ambiguities. Our initial analysis on the verb-noun ambiguity using the stimuli by Aina and Linzen (2021) (like the example in Figure 1) led to inconclusive results. For such cases, maybe other metrics are required, especially because it is harder to isolate the effect of the ambiguity from the effect that the immediate next token has on its neighbour. Further investigation can also be done with ungrammatical sentences and other types of garden paths in the SAP benchmark.

As we discuss in the Appendix, some stimuli were excluded from the analysis of meaning due to rare tokens that require subtokenisation and cause misalignment in the incremental structures.

For the analysis of causal embeddings, we used different architectures. Ideally, we should also compare a fixed architecture, trained with all the same parameters and data, but once with causal masking and once with bidirectional access. That way we could directly compare how the RI bidirectional and causal embeddings differ.

We do not perform layer-level analysis for dependency parsing as both parsers use scalar mix of hidden representations from pre-trained LMs. While we only use graph-based dependency parsers, it would be interesting to apply our methods to transition-based parsers, as they works from left to right and are traditionally associated with the notion of incrementality (Nivre, 2008). In our preliminary analysis, we also compare Shannon's entropy on attention distributions for dependency arcs. However, the results are inconclusive.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Laura Aina and Tal Linzen. 2021. The language model understood the prompt was ambiguous: Probing syntactic uncertainty through generation. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 42–57, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 301–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. 2021. Biaffine dependency and semantic graph parsing for EnhancedUniversal dependencies. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 184–188, Online. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Niels Beuck, Arne Köhn, and Wolfgang Menzel. 2011. Decision strategies for incremental POS tagging. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 26–33, Riga, Latvia. Northern European Association for Language Technology (NEALT).

Thomas Bever. 1970. *The Cognitive Basis for Linguistic Structures*, pages 279–352.

Idan A. Blank. 2023. What are large language models supposed to model? *Trends in Cognitive Sciences*, 27(11):987–989.

Angelica Chen, Ravid Schwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2023. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. *arXiv preprint arXiv:2309.07311*.

Angelica Chen, Vicky Zayats, Daniel Walker, and Dirk Padfield. 2022. Teaching BERT to wait: Balancing accuracy and latency for streaming disfluency detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 827–838, Seattle, United States. Association for Computational Linguistics.

Yeong-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations. *arXiv preprint arXiv:2303.09435*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. 2022. Probing for incremental parse states in autoregressive language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2801–2813, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Lyn Frazier. 1979. *On comprehending sentences: Syntactic parsing strategies*. Ph.D. thesis, University of Connecticut.

Lyn Frazier and Keith Rayner. 1982. Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14(2):178–210.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Honnibal and Mark Johnson. 2014. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, 2:131–142.

Lianna Hrycyk, Alessandra Zarcone, and Luzian Hahn. 2021. Not so fast, classifier – accuracy and entropy reduction in incremental intent classification. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 52–67, Online. Association for Computational Linguistics.

Kuan-Jung Huang, Suhas Arehalli, Mari Kugemoto, Christian Muxica, Grusha Prasad, Brian Dillon, and

Tal Linzen. 2023. Surprisal does not explain syntactic disambiguation difficulty: evidence from a large-scale benchmark.

Tovah Irwin, Kyra Wilson, and Alec Marantz. 2023. BERT shows garden path effects. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3220–3232, Dubrovnik, Croatia. Association for Computational Linguistics.

William Jurayj, William Rudman, and Carsten Eickhoff. 2022. Garden path traversal in GPT-2. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 305–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Patrick Kahardipraja, Brielen Madureira, and David Schlangen. 2021. Towards incremental transformers: An empirical analysis of transformer models for incremental NLU. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1178–1189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Kahardipraja, Brielen Madureira, and David Schlangen. 2023. TAPIR: Learning adaptive revision for incremental natural language understanding with a two-pass model. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4173–4197, Toronto, Canada. Association for Computational Linguistics.

Ayush Kaushal, Aditya Gupta, Shyam Upadhyay, and Manaal Faruqui. 2023. Efficient encoders for streaming sequence tagging. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 418–429, Dubrovnik, Croatia. Association for Computational Linguistics.

Casey Kennington, Spyros Kousidis, and David Schlangen. 2014. InproTKs: A toolkit for incremental situated processing. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 84–88, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Hatim Khouzaimi, Romain Laroche, and Fabrice Lefevre. 2014. An easy method to make dialogue systems incremental. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 98–107, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Anne Kilger and Wolfgang Finkler. 1995. Incremental generation for real-time applications. *DFKI Verbmobil Research Report RR-95-11*.

Nikita Kitaev, Thomas Lu, and Dan Klein. 2022. Learned incremental representations for parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3086–3095, Dublin, Ireland. Association for Computational Linguistics.

Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1):167–207.

Jonghyun Lee and Jeong-Ah Shin. 2023. Decoding bert's internal processing of garden-path structures through attention maps. *Korean Journal of English Language and Linguistics*, 23:461–481.

Jonghyun Lee, Jeong-Ah Shin, and Myung-Kwan Park. 2022. (al)bert down the garden path: Psycholinguistic experiments for pre-trained language models. *Korean Journal of English Language and Linguistics*, 22:1033–1050.

Willem JM Levelt. 1993. *Speaking: From intention to articulation*. MIT press.

Alma Lindborg and Milena Rabovsky. 2021. Meaning in brains and machines: Internal activation update in large-scale language model partially reflects the n400 brain potential. In *Proceedings of the annual meeting of the cognitive science society*, volume 43.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. 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. 2019b. Roberta: A robustly optimized bert pretraining approach.

Brielen Madureira, Patrick Kahardipraja, and David Schlangen. 2023. The road to quality is paved with good revisions: A detailed evaluation methodology for revision policies in incremental sequence labelling. In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 156–167, Prague, Czechia. Association for Computational Linguistics.

Brielen Madureira and David Schlangen. 2020. Incremental processing in the age of non-incremental encoders: An empirical assessment of bidirectional models for incremental NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 357–374, Online. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

William Marslen-Wilson. 1973. Linguistic structure and speech shadowing at very short latencies. *Nature*, 244(5417):522–523.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560, Singapore. Association for Computational Linguistics.

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

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Andrew Rafla and Casey Kennington. 2019. Incrementalizing rasa's open-source natural language understanding pipeline. *arXiv preprint arXiv:1907.05403*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Koichiro Ryu, Shigeki Matsubara, and Yasuyoshi Inagaki. 2006. Simultaneous English-Japanese spoken language translation based on incremental dependency parsing and transfer. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 683–690, Sydney, Australia. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

David Schlangen. 2023. On general language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8818–8825, Singapore. Association for Computational Linguistics.

David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.

Sukanta Sen, Rico Sennrich, Biao Zhang, and Barry Haddow. 2023. Self-training reduces flicker in retranslation-based simultaneous translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3734–3744, Dubrovnik, Croatia. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Drahomíra Spoustová and Miroslav Spousta. 2010. Dependency parsing as a sequence labeling task. *The Prague Bulletin of Mathematical Linguistics*, 94.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. 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 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Dennis Ulmer, Dieuwke Hupkes, and Elia Bruni. 2019. Assessing incrementality in sequence-to-sequence models. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 209–217, Florence, Italy. Association for Computational Linguistics.

Marten Van Schijndel and Tal Linzen. 2021. Single-stage prediction models do not explain the magnitude of syntactic disambiguation difficulty. *Cognitive science*, 45(6):e12988.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Davis Yoshida and Kevin Gimpel. 2021. Reconsidering the past: Optimizing hidden states in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4099–4105, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Appendix

In this section, we provide some additional details and results.

Classic garden paths The NP/S garden path was originally discussed by Frazier (1979) and the MVRR by Bever (1970). We did not use the other forms of garden path in the SAP benchmark (Huang et al., 2023) because the type revision would be harder to isolate or, in the case of NP/Z and subject-verb agreement mismatch, the stimulus is not a completely well-formed construction in written text.

NNC The template we selected for the stimuli is `This is a noun1 noun2`, and for the baseline reference is `This is a noun1 ,` . This was chosen to be a not very informative left context in order to focus on the effect of the NP construction.

Licenses The NNC stimuli[6] are released without a license. The NP/S and MVRR stimuli[7] and the repository as a whole is under the MIT license. BERT and ELECTRA are released under Apache 2.0. RoBERTa and GPT-2 are under MIT. OPT is under a custom OPT-175B license agreement. The dependency parsing libraries we used are under the MIT license.

B Details: Incremental Construction of Meaning

We use the pretrained model checkpoints and corresponding tokenizers available on HuggingFace: `bert-base-uncased`,[8] `roberta-base`,[9] `gpt2`,[10] and `facebook/opt-125m`.[11]

[6] https://github.com/marcospln/noun_compound_senses

[7] https://github.com/caplabnyu/sapbenchmark

[8] https://huggingface.co/bert-base-uncased

[9] https://huggingface.co/roberta-base

[10] https://huggingface.co/gpt2

[11] https://huggingface.co/facebook/opt-125m

We did not include the stimuli for which tokens were split into subtokens, because that creates misalignment in the triangular structures and thus requires workarounds. Because of that, the samples differed slightly for each model. This is not a problem in our analysis because we are not ranking the performance of the models; what matters is the intrinsic behaviour of each model separately. The number of excluded instances is shown in Table 3.

model	source	n excluded
BERT	NP/S	5
BERT	MVRR	4
BERT	NNC	14
RoBERTa	NP/S	1
RoBERTa	MVRR	0
RoBERTa	NNC	23
opt	NP/S	1
opt	MVRR	0
opt	NNC	23
gpt2	NP/S	1
gpt2	MVRR	0
gpt2	NNC	23

Table 3: Number of excluded instances for each model and type of stimuli due to subtokenisation.

Cosine distance was chosen because that should capture meaning variations in the embedding space. It is also possible to use other distance metrics like Manhattan or Euclidean distance. We used the `paired_distances` method from the `sklearn.metrics.pairwise` package[12] to compute the cosine distance.

B.1 Computations

We now describe in greater detail the computation steps we took to derive the plots the results they illustrate.

Each sentence was fed to each model prefix by prefix and after each step we saved the hidden representations. We stored the representations of all layers using 4D arrays with dimensions number of layers, sequence length (time step), sequence length (token position), embedding dimension. All models had an initial embedding layer and 12 encoding layers.

NNC We extracted the states for all stimuli and their corresponding baseline. Then, we computed the cosine distance of each state to its own version in the preceding time step. For the main diagonal

[12] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.paired_distances.html#sklearn.metrics.pairwise.paired_distances

(*i.e.* the first time a token's state is constructed), we set this value to be 0 for a better visualisation. This resulted in a 2D lower diagonal matrix where each cell contains one value. Let c_s and c_b be these charts for the stimulus and its baseline, respectively. c_s has one row and one column more than c_b, since the baseline does not contain the second noun and is thus one token shorter. We create c'_s by deleting the last row and last column in c_s (which correspond to the comma) and then compute the absolute difference $d = |c'_s - c_b|$. We then average the numbers in d over all stimulus+baseline pairs for each layer.

NP/S and MVRR We extracted the states for all stimuli and their corresponding baseline. Then, we computed the cosine distance of each state to its own version in the last time step, which represents its meaning given the full sentence as context. This resulted in a 2D lower diagonal matrix where each cell contains one value. Let c_s and c_b be these charts for the stimulus and its baseline, respectively. Here, c_b has more rows/columns due to the added tokens that disambiguate the verb (*i.e. that* or *who was*). We create c'_b by deleting the extra row(s) and column(s) in c_b (which correspond to the added tokens) and then compute the absolute difference $d = |c_s - c'_b|$. We then average the numbers in d over all stimulus+baseline pairs for each layer. Some sentences start with a `det adj noun` and other with `det noun`. For the plots, we remove the initial rows/columns and begin the chart at the aligned noun.

Causal For the analysis of causal embeddings, we use four variations of each sentence: (a) the stimulus with a temporary ambiguity; (b) the baseline, which disambiguates the role of the verb and NP in advance; (c) the stimulus with the first verb replaced with an unambiguous verb; (d) the baseline with the first verb replaced by the same unambiguous verb. For NP/S, we use the verb *said* and for MVRR, *given*. For example, for NP/S:

(a) The new doctor demonstrated the operation appeared increasingly likely to succeed.

(b) The new doctor demonstrated that the operation appeared increasingly likely to succeed.

(c) The new doctor said the operation appeared increasingly likely to succeed.

(d) The new doctor said that the operation appeared increasingly likely to succeed.

and for MVRR:

(a) The professor awarded the grant gained more attention from marine biologists.

(b) The professor who was awarded the grant gained more attention from marine biologists.

(c) The professor given the grant gained more attention from marine biologists.

(d) The professor who was given the grant gained more attention from marine biologists.

We first compute the distance of the states of the tokens in (a) to their corresponding states in (b) (*i.e.* the states of the added tokens are ignored). Let us call the resulting vector d_{ab}. To account for the expected variation due to the different number of tokens, we do the same for (c) and (d), to be used as a reference, and get d_{cd}. Then, we take the absolute difference $|d_{ab} - d_{cd}|$ and average values over all stimuli.

To conclude, we detail the procedure to compute the numbers in Table 1. We first create triangular structures for all baseline sentences by computing the cosine distance of each state to its own version in the immediately preceding time step. The cell at row i and column j in the right triangle thus contain a value representing how much token w_i has affected the state s_j of token w_j, with $i > j$. We then average the distances for all tokens that have a corresponding state $1, 2, \ldots, 7$ steps into the future. In practice, this means collecting values at each sub-diagonal $-1, -2, \ldots, -7$ and computing the average over baseline stimuli.

B.2 Additional Results

In BERT, for layers 1 to 10, the maximum average change occurs at the first noun when the second noun is observed. For the upper layers, it is the second most, but still with a mean distance of almost 0.41, considerably above the corresponding 0.34 in Table 1. In the full overview in Figure 17 we see that all previous tokens are affected by all right tokens, especially during the construction of the NP. In Figure 10, we see that RoBERTa has a similar effect on the NNC stimuli as BERT: the second noun influences the whole prefix, especially the first noun. Here, however, the magnitude of the effect is smaller, roughly half that of BERT. Besides, the upper layers have effects as high as the middle layers, different from BERT, where two of the middle layers stand out. Figure 20 shows again

that the magnitude of the state distances is smaller for RoBERTa. Up to layer 7, the effect of the second noun on the first one is the largest. However, in the upper layers, this no longer holds: The largest variation occurs for *this* when *is* is observed.

Figures 18 and 19 show the full results of the NP/S and MVRR cases for BERT. For RoBERTa, the patterns of behaviour are similar: The representation of the first verb differs from what will be the final one, until the second verb, and one token after that, are integrated. Again, while BERT has an effect of around 0.15, RoBERTa's effect is around 0.05, as illustrated in Figure 21 and 22. In both types of ambiguity, the effect becomes smaller in the last layer for RoBERTa, while it persists in BERT. More investigation is needed to shed light on what conceptual differences between the models cause the different behaviours.

Figure 11 shows the analysis of the causal embeddings for OPT. Like GPT-2, it creates dissimilar representations for the ambiguous and the unambiguous prefixes, but here the effect last until the final layer, *i.e.* it will influence the subsequent predictions. Figures 13 and 12 show directly the distance between stimulus and baseline, without subtracting the counterpart pair. This is to show that, indeed, for GPT-2, the representations of the last layer are very similar, despite the disambiguation in advance.

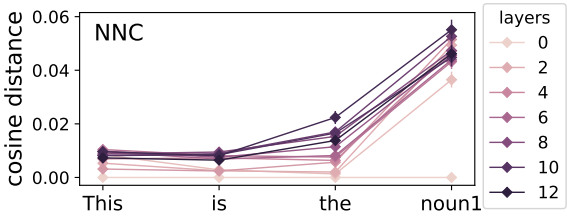


Figure 10: **NNC**. Average (absolute) effect over a baseline of the second noun on the tokens in the prefix for RoBERTa.

C Details: Incremental Dependency Parsing

We use the biaffine parser implemented in `SuPar` (`https://github.com/yzhangcs/parser`) with embeddings from RoBERTa large. For the DiaParser, we follow the original implementation (`https://github.com/Unipisa/diaparser`) using ELECTRA base. Both parsers use `stanza` (Qi et al., 2020) as tokenizers.

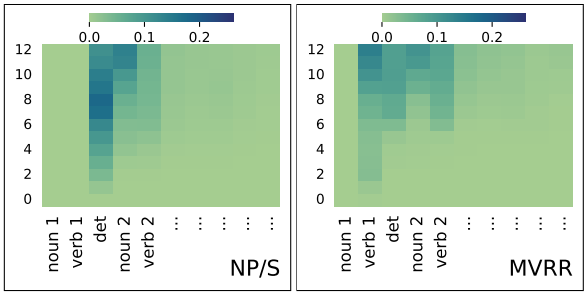


Figure 11: Distance between the causal embeddings of the stimulus and the baseline, after subtracting the expected variation by a counterpart pair with an unambiguous verb, for all layers of OPT.

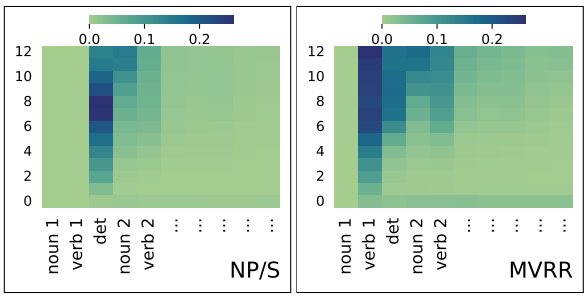


Figure 12: Distance between the causal embeddings of the stimulus and the baseline, for all layers of OPT.

C.1 Computation

Our method is inspired by the approach of Hrycyk et al. (2021). Let us consider $\mathsf{head}(i)^t$ as the head of token i predicted at time step t. When $\mathsf{head}(i)^t = \mathsf{head}(i)^{ref}$, we can directly compare the label attention distribution between the current time step t against the reference time step $ref \in \{0, t-1, T\}$. However when $\mathsf{head}(i)^t \neq \mathsf{head}(i)^{ref}$, either heads may or may not be observed yet. In the case where the head is already observed, we use the label distribution from the self-attention matrix at time step t or ref for comparison depending whether $t < ref$, otherwise we assume a uniform distribution. To be more precise, let assume that in an incremental scenario, the attention distribution for dependency labels at time step n is also available at time step m where $m > n$. We then compute the Jensen-Shannon divergence as the following when $\mathsf{head}(i)^n \neq \mathsf{head}(i)^m$:

$$JSD(p(y|\mathsf{arc}(i,j)^n)||p(y|\overline{\mathsf{arc}}(i,j)^m)) \quad (1)$$

$$JSD(p(y|\overline{\mathsf{arc}}(i,k)^n)||p(y|\mathsf{arc}(i,k)^m)) \quad (2)$$

where $\mathsf{arc}(i,j)$ is the arc between token i and j, and $p(y|\overline{\mathsf{arc}}(i,\cdot))$ is obtained from the self-attention matrix if the token is already observed and uniform distribution otherwise. The distance

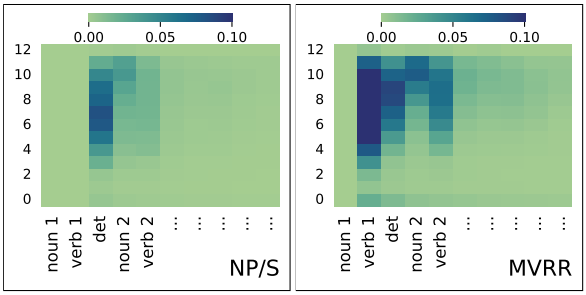


Figure 13: Distance between the causal embeddings of the stimulus and the baseline, for all layers of GPT-2.

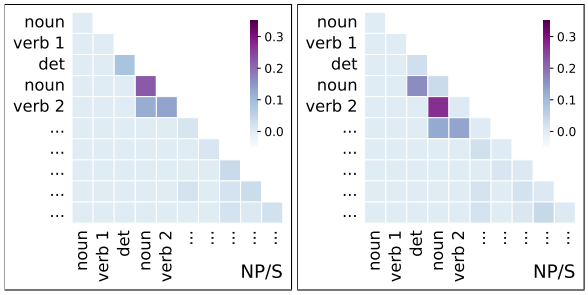
between the label attention distribution at time step n and m is then defined as the average of (1) and (2).

C.2 Additional Results

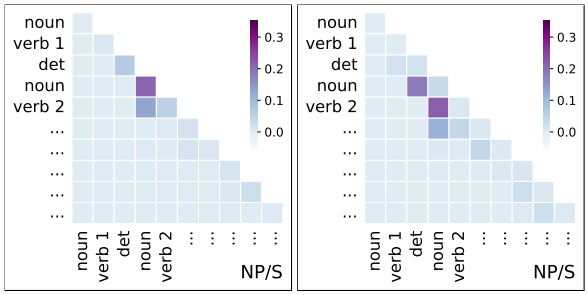
NP/S For the last time step as a reference, we see that the complement's distribution is closer to the final for the baseline compared to the stimulus, showing that the complementiser helps in processing the temporary ambiguity. The overview is depicted in Figure 24. We also measure how the label distribution changes as more token is observed by taking the absolute difference of JSD between the stimulus and the baseline wrt. the previous time step. In Figure 14 (right), we observe a noticeable difference in the divergence of the complement when the disambiguation token (the second verb) is encountered. After that point, the gap diminishes rapidly.

MVRR We compute the absolute difference of JSD wrt. the last step as the stimulus and the baseline have the same final interpretation. In Figure 15 (left), we observe that the first noun and the first verb are processed differently in both cases. This is highly likely due to the fact that the first verb can be interpreted as the main verb or as a reduced relative, which also affects how the first noun is understood. We see that both the stimulus and the baseline converge to the same interpretation after the disambiguation token (the second verb) is observed.

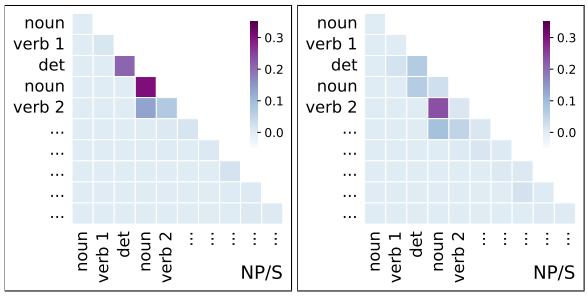
Alignment In addition to MCC, we also compute the average precision (AP) score for both the stimulus and the baseline. We do this by treating edits as the ground truth and the JSD wrt. previous step as the prediction to see if edits can be predicted just from JSD alone (Table 4). We observe that the AP for dependency arcs are high in general, while it is lower for the labels. We also see that the average edit ratio is low for all stimuli except NNC, as shown in Table 5.



(a) Biaffine parser



(b) DiaParser (ELECTRA-EWT)

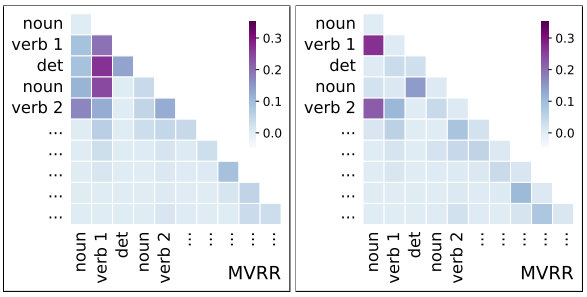


(c) DiaParser (ELECTRA-PTB)

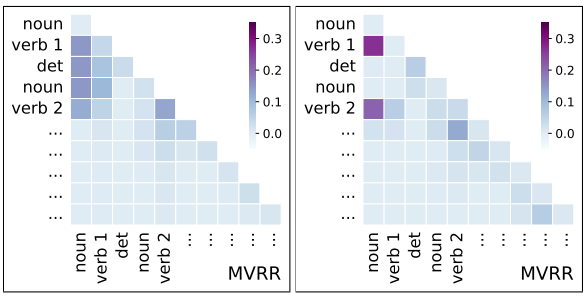
Figure 14: Average absolute difference of JSD between stimulus and baseline on NP/S. Left: last step as reference. Right: previous step as reference. We also include the results for biaffine parser for the ease of comparison.

	NNC		NP/S		MVRR	
Arc	S	B	S	B	S	B
Biaffine	0.99	0.99	0.92	0.88	0.88	0.89
DiaParser (EWT)	0.99	1.00	0.98	0.96	0.91	0.97
DiaParser (PTB)	0.99	0.99	0.97	0.98	0.92	0.99
Label	S	B	S	B	S	B
Biaffine	0.41	0.90	0.92	0.78	0.80	0.85
DiaParser (EWT)	0.30	0.48	0.86	0.76	0.75	0.87
DiaParser (PTB)	0.55	0.80	0.89	0.74	0.78	0.91

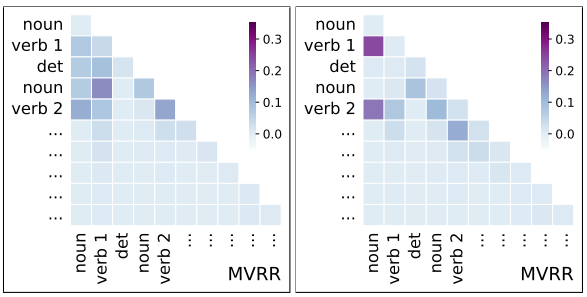
Table 4: Average Precision between JSD wrt. previous time step and edits for dependency arcs and labels. S: stimulus & B: baseline.



(a) Biaffine parser

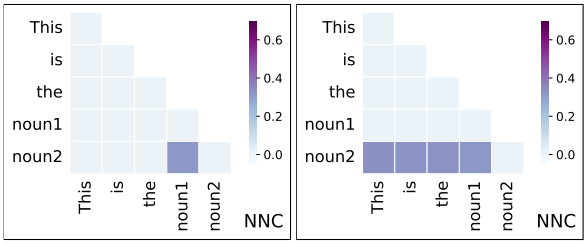


(b) DiaParser (ELECTRA-EWT)

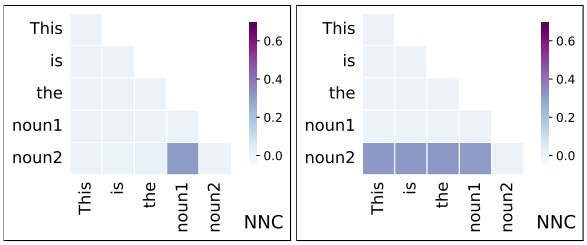


(c) DiaParser (ELECTRA-PTB)

Figure 15: Average absolute difference of JSD between stimulus and baseline on MVRR. Left: last step as reference. Right: previous step as reference. We also include the results for biaffine parser for the ease of comparison.



(a) DiaParser (ELECTRA-EWT)



(b) DiaParser (ELECTRA-PTB)

Figure 16: Average absolute difference of JSD between stimulus and baseline on NNC. Left: first step as reference. Right: previous step as reference.

	NNC		NP/S		MVRR	
Arc	S	B	S	B	S	B
Biaffine	0.53	0.41	0.11	0.11	0.12	0.10
DiaParser (EWT)	0.67	0.60	0.13	0.13	0.15	0.11
DiaParser (PTB)	0.53	0.40	0.11	0.12	0.12	0.09
Label	S	B	S	B	S	B
Biaffine	0.27	0.31	0.09	0.08	0.09	0.09
DiaParser (EWT)	0.27	0.30	0.09	0.08	0.11	0.08
DiaParser (PTB)	0.27	0.30	0.09	0.08	0.10	0.08

Table 5: Average edit ratio for dependency arcs and labels. S: stimulus & B: baseline.

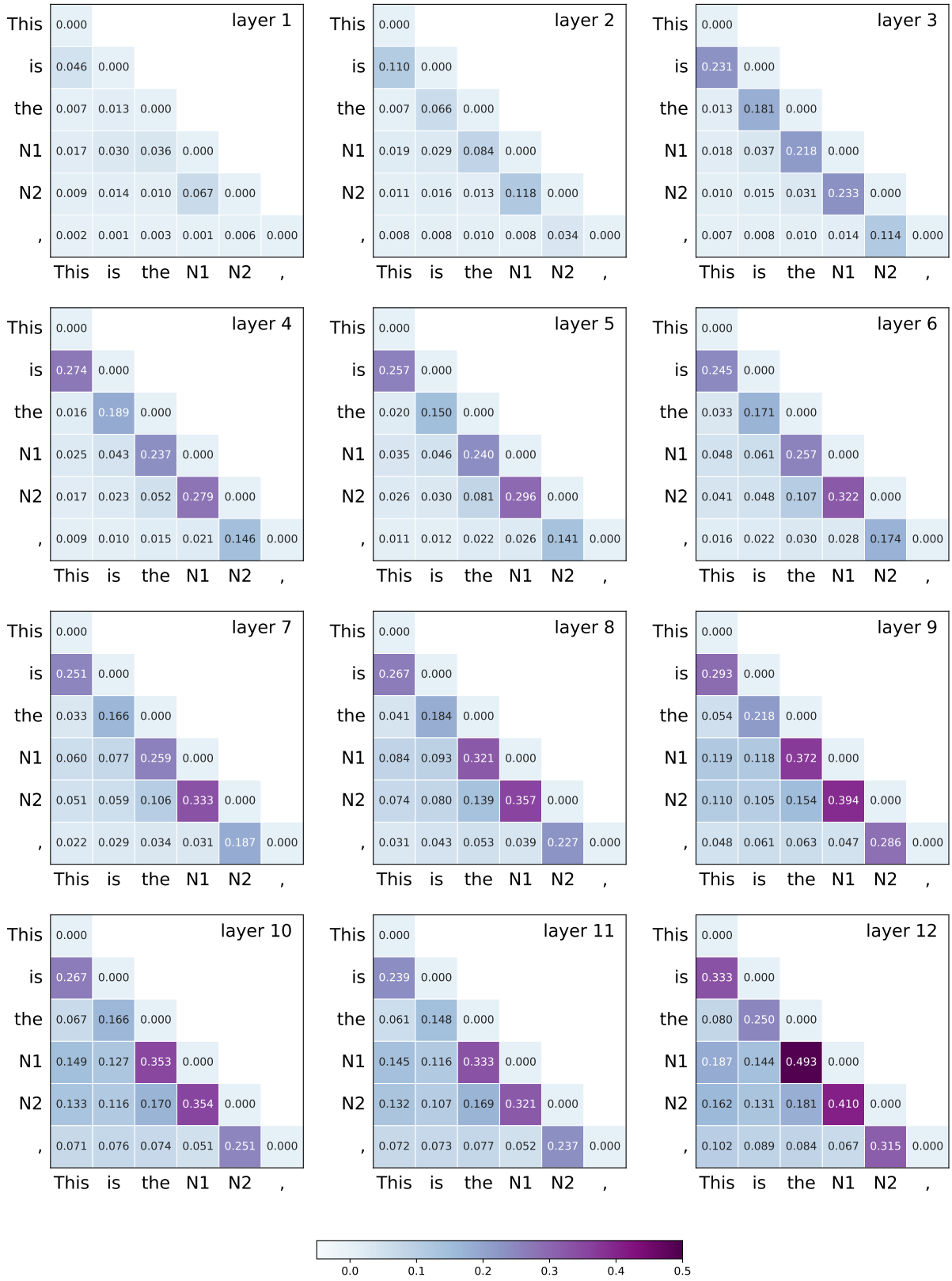


Figure 17: Overview of the mean cosine distance of a contextual embedding to its state in the preceding time step, for all layers of BERT, averaged over all NNC stimuli.

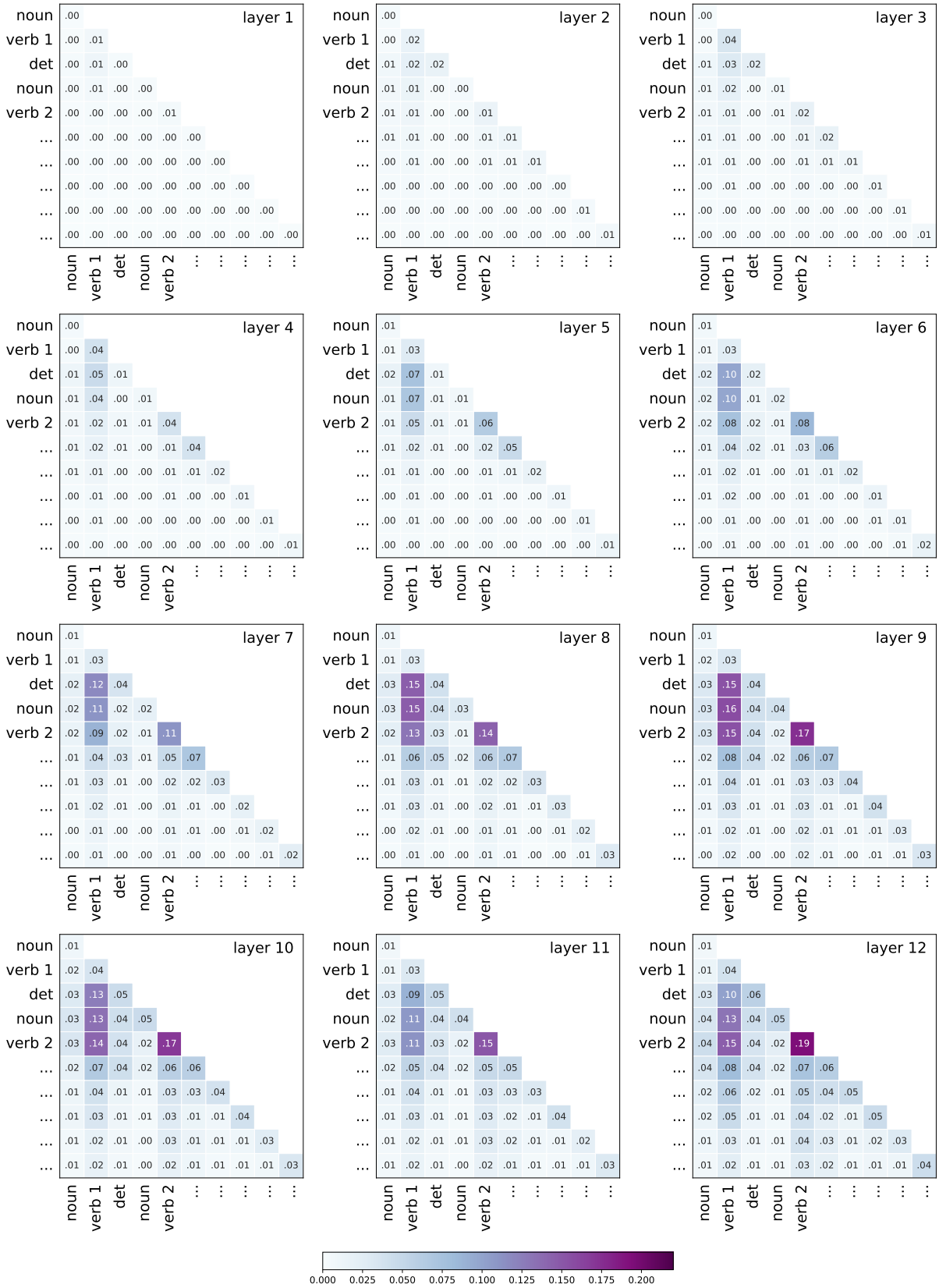


Figure 18: Overview of the cosine distance of a contextual embedding to its final state (last row). The numbers represent the absolute difference over the unambiguous baseline, for all layers of BERT, averaged over all NP/S stimuli.

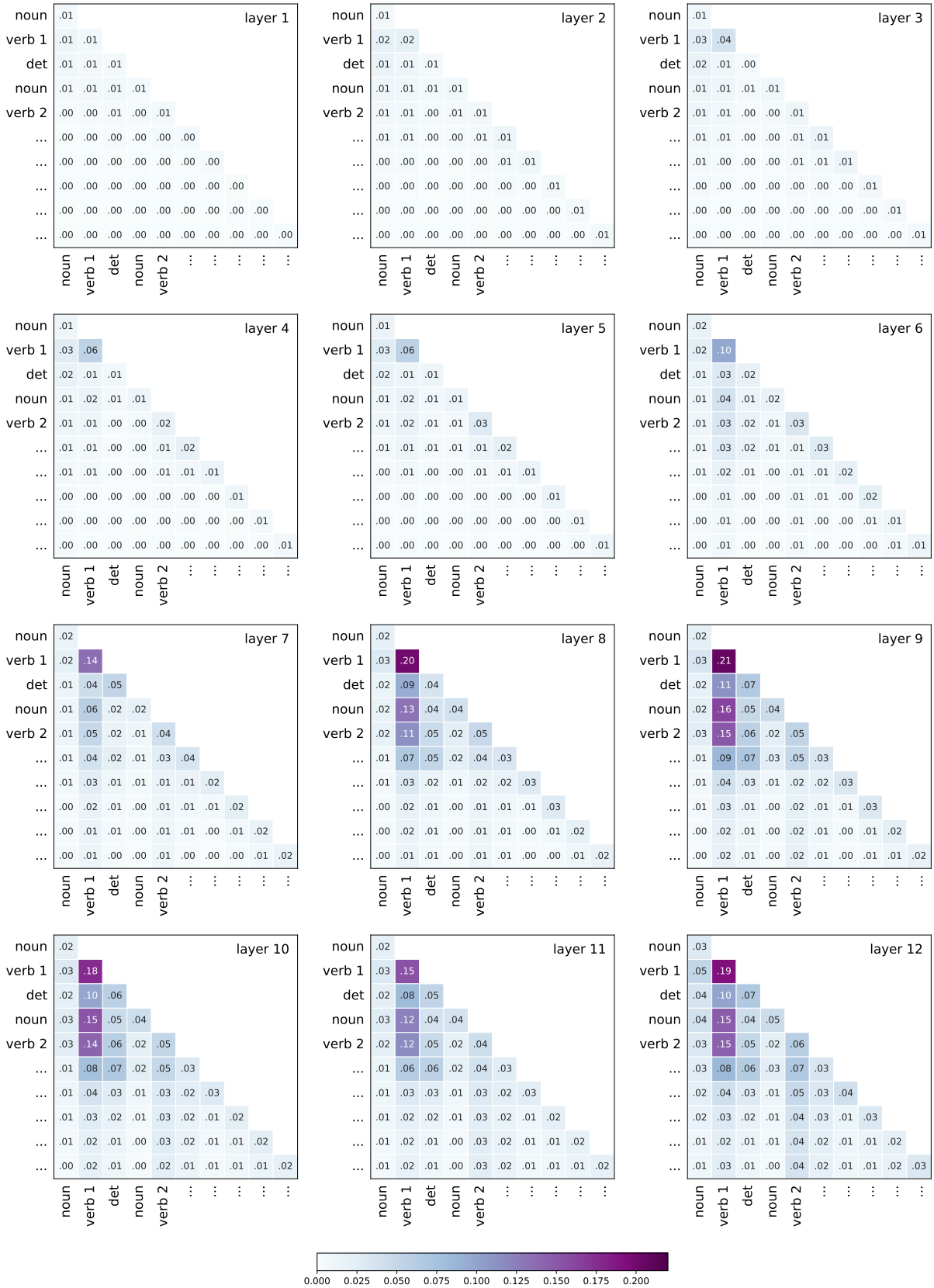


Figure 19: Overview of the cosine distance of a contextual embedding to its final state (last row). The numbers represent the absolute difference over the unambiguous baseline, for all layers of BERT, averaged over all MVRR stimuli.

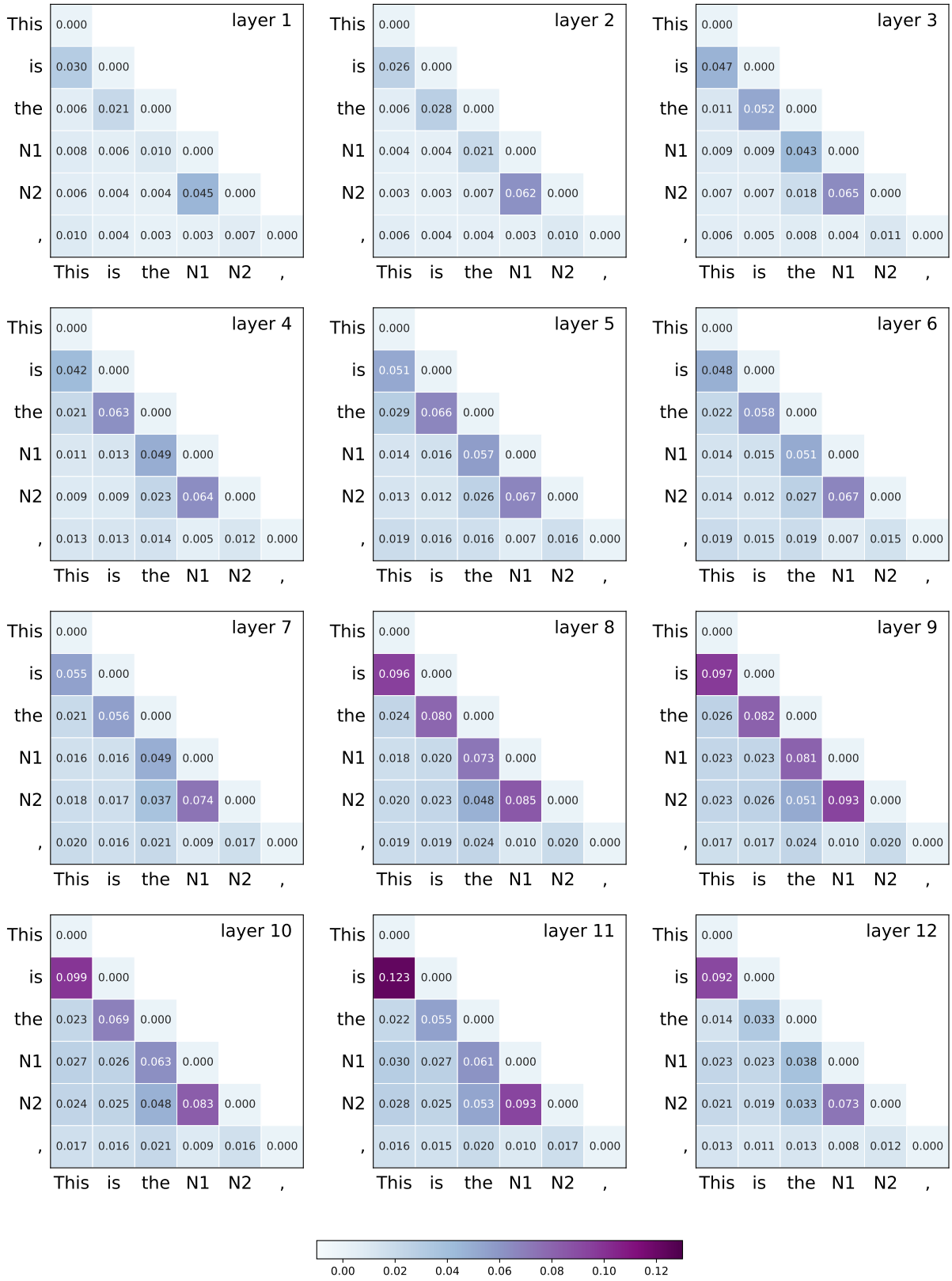


Figure 20: Overview of the mean cosine distance of a contextual embedding to its state in the preceding time step, for all layers of RoBERTa, averaged over all NNC stimuli.

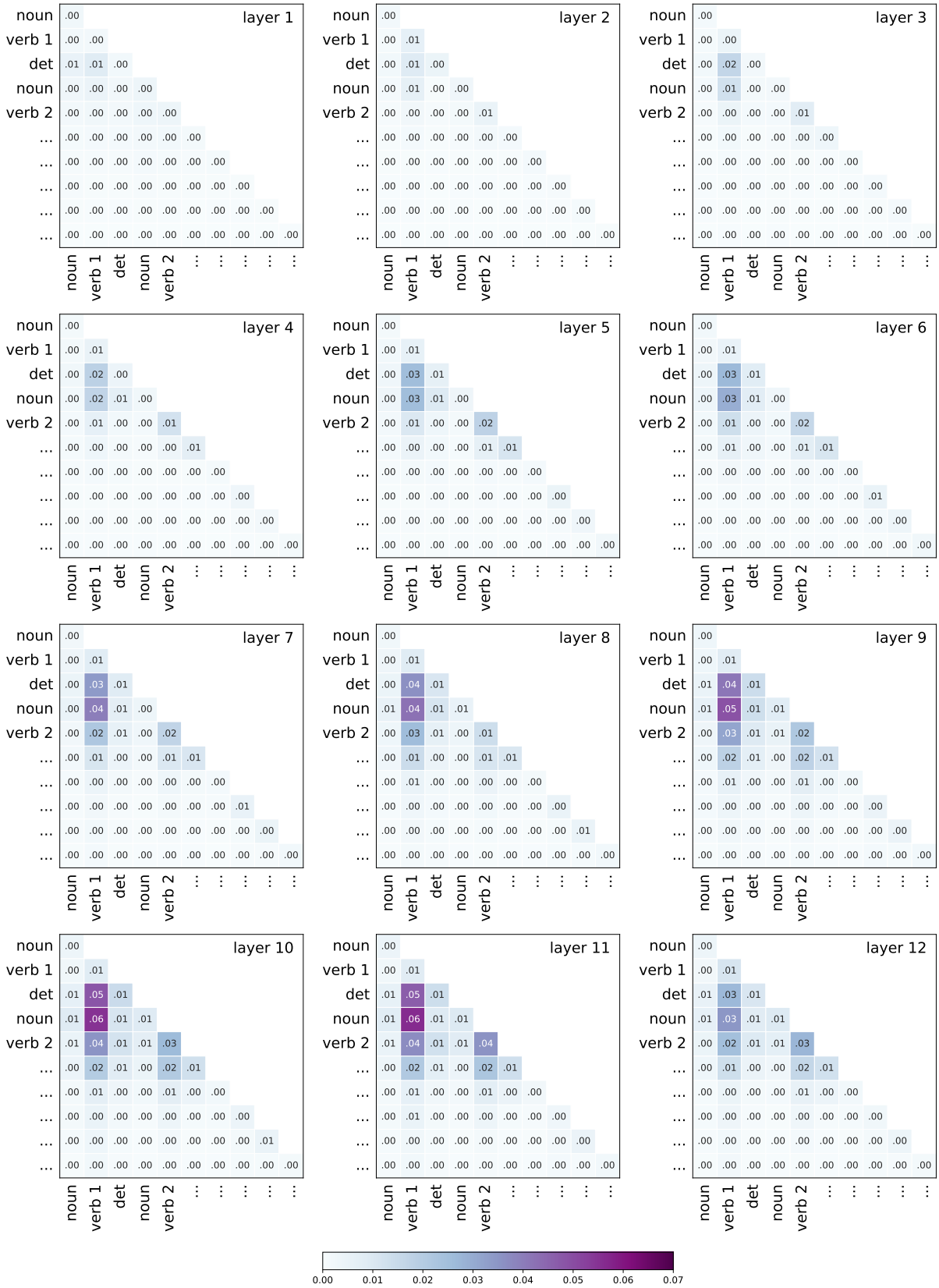


Figure 21: Overview of the cosine distance of a contextual embedding to its final state (last row). The numbers represent the absolute difference over the unambiguous baseline, for all layers of RoBERTa, averaged over all NP/S stimuli.

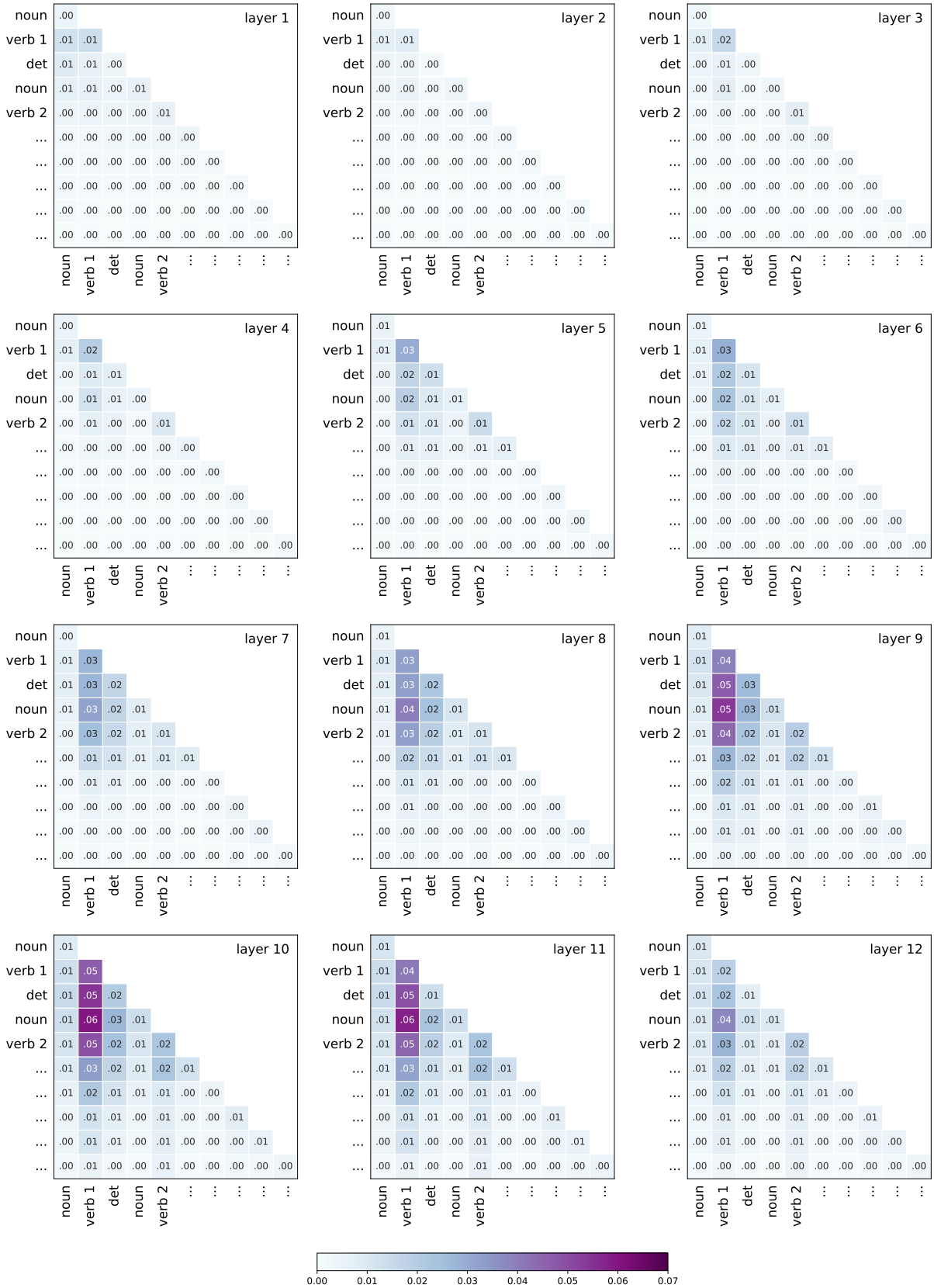
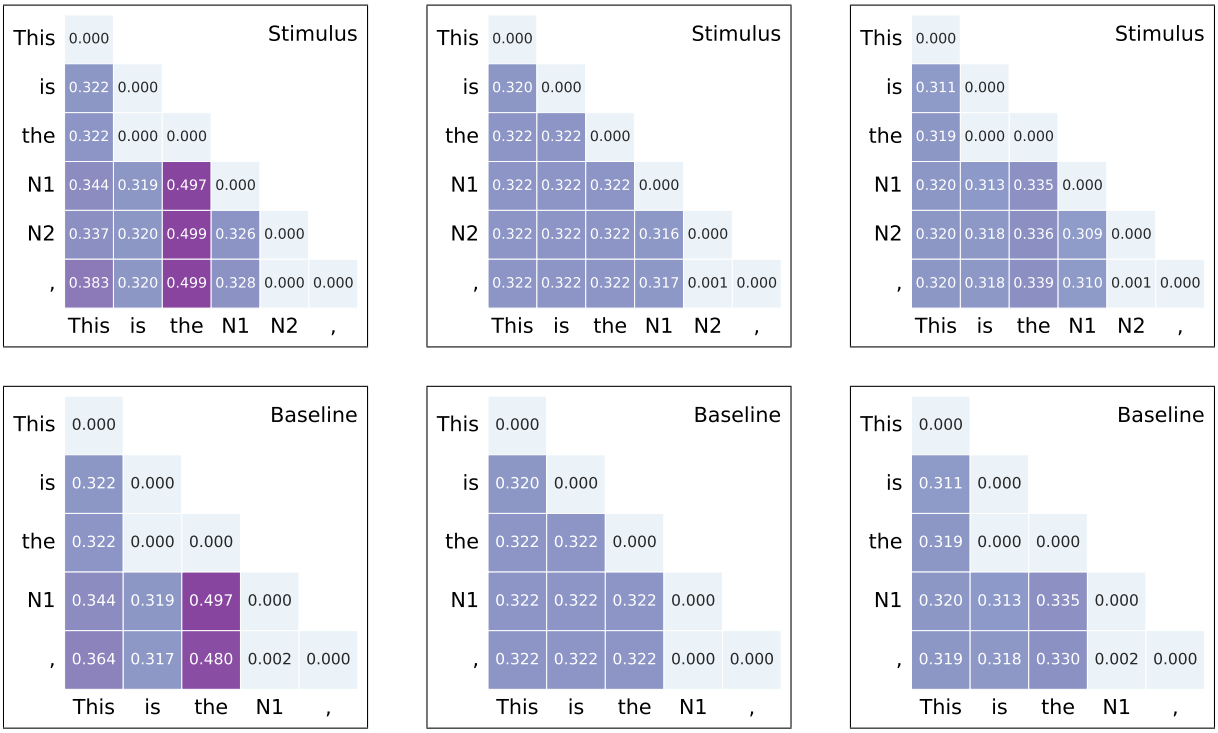
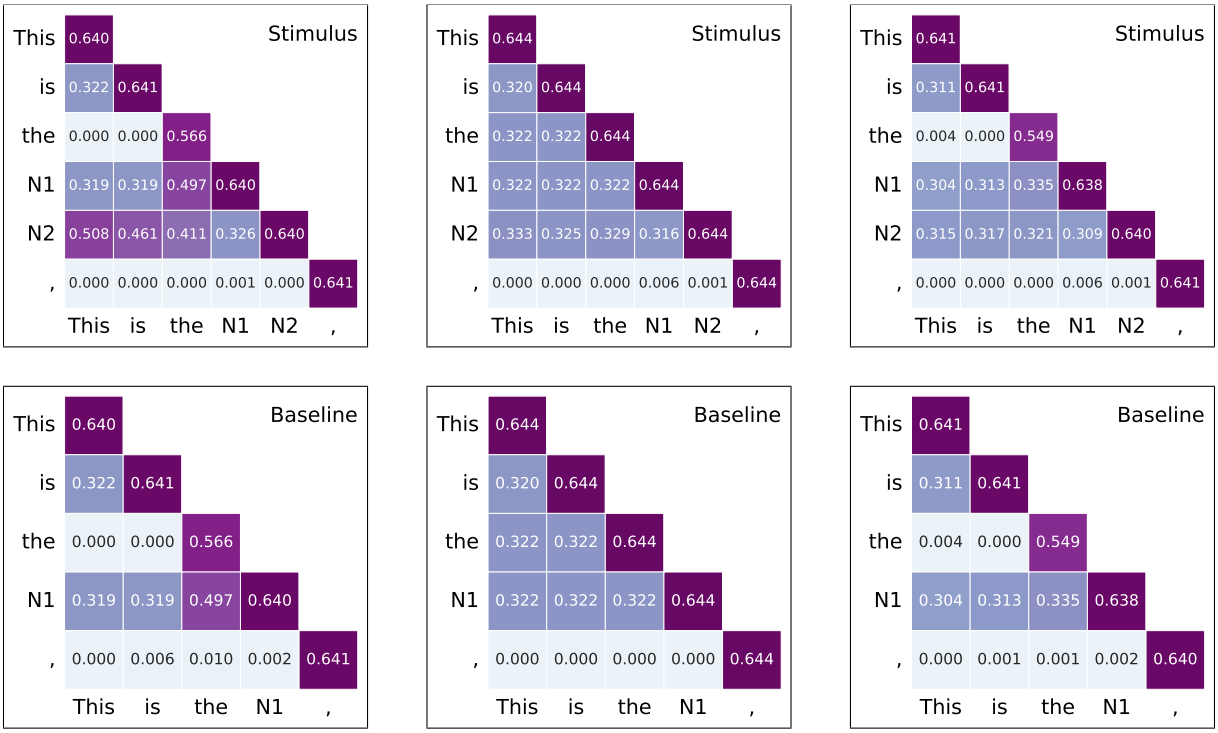


Figure 22: Overview of the cosine distance of a contextual embedding to its final state (last row). The numbers represent the absolute difference over the unambiguous baseline, for all layers of RoBERTa, averaged over all MVRR stimuli.

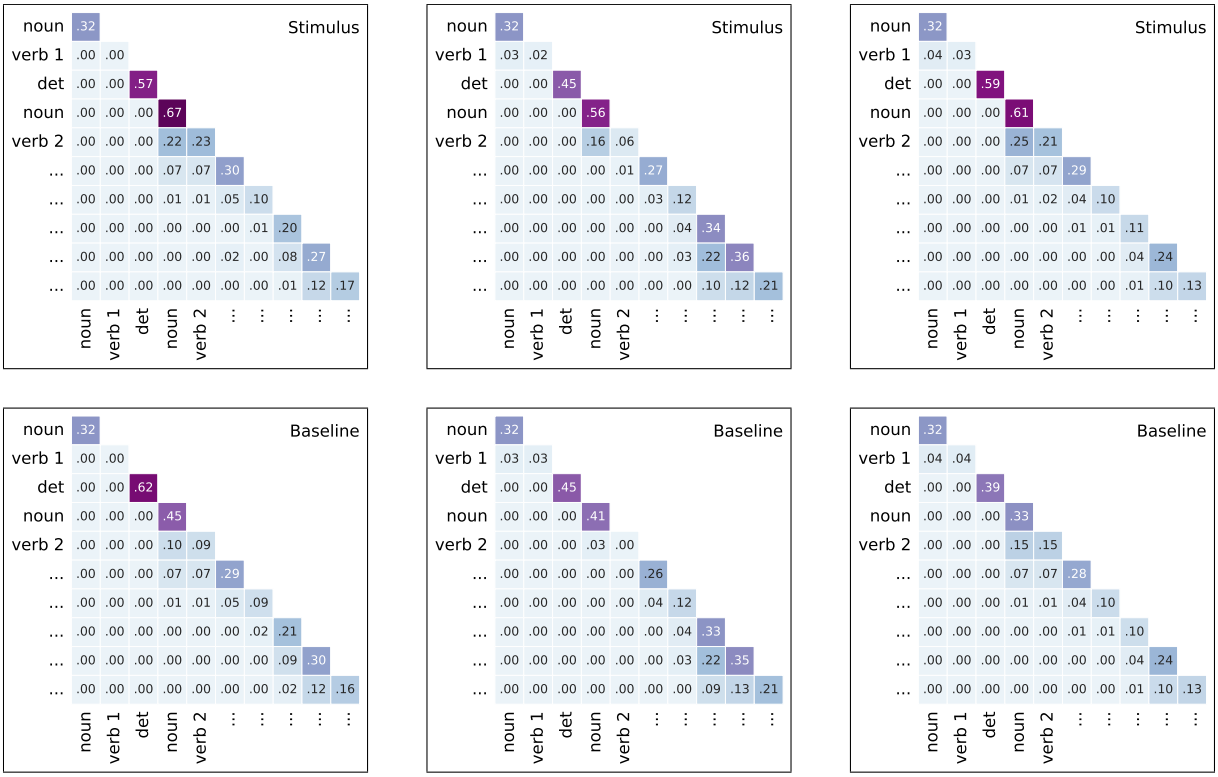


(a) JS divergence wrt. the first time step

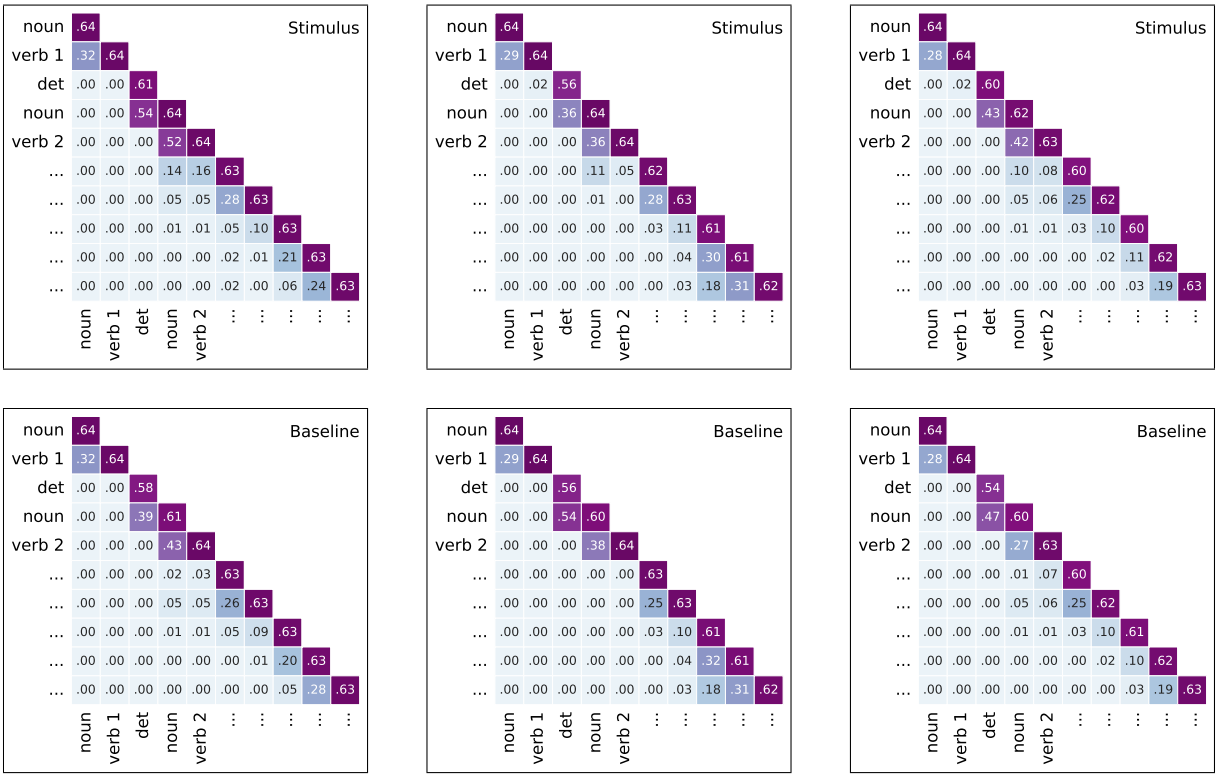


(b) JS divergence wrt. the previous time step

Figure 23: Overview of the JSD for the label attention distributions wrt. the first and previous time steps as reference, averaged over all NNC stimuli and baselines. From left to right: biaffine parser, DiaParser (ELECTRA-EWT and ELECTRA-PTB).

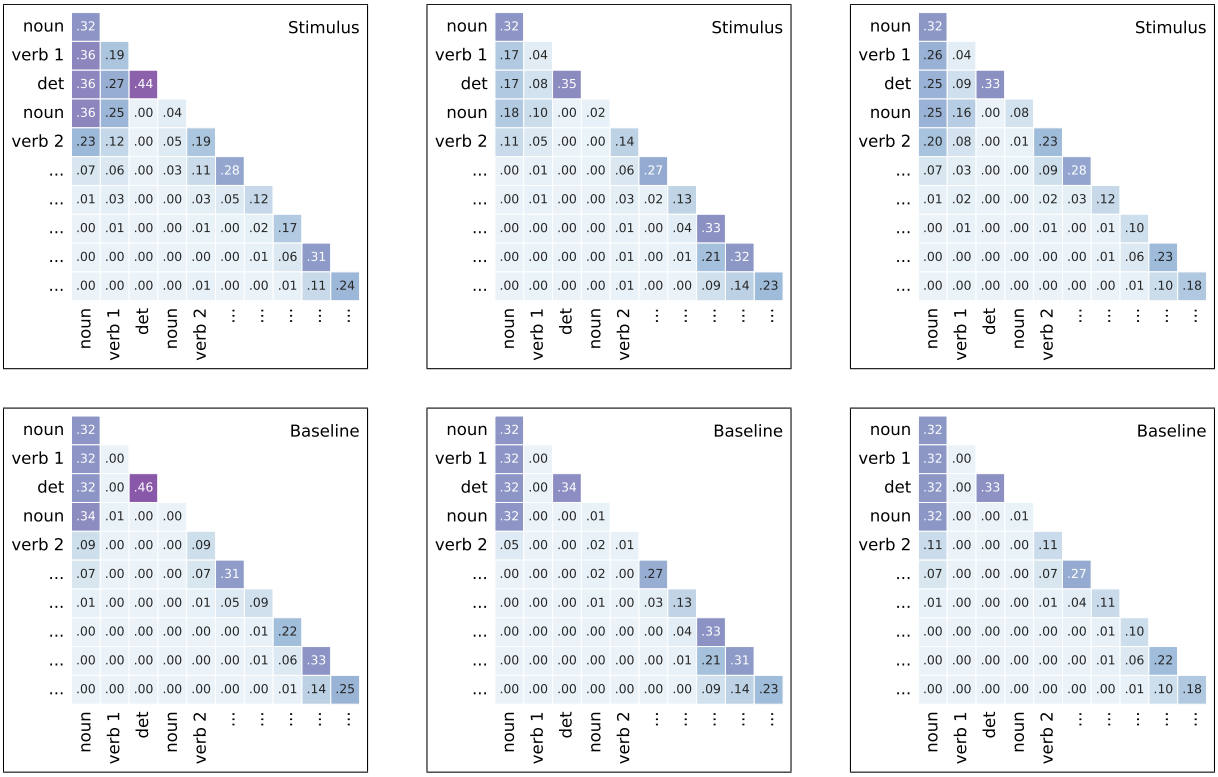


(a) JS divergence wrt. the last time step

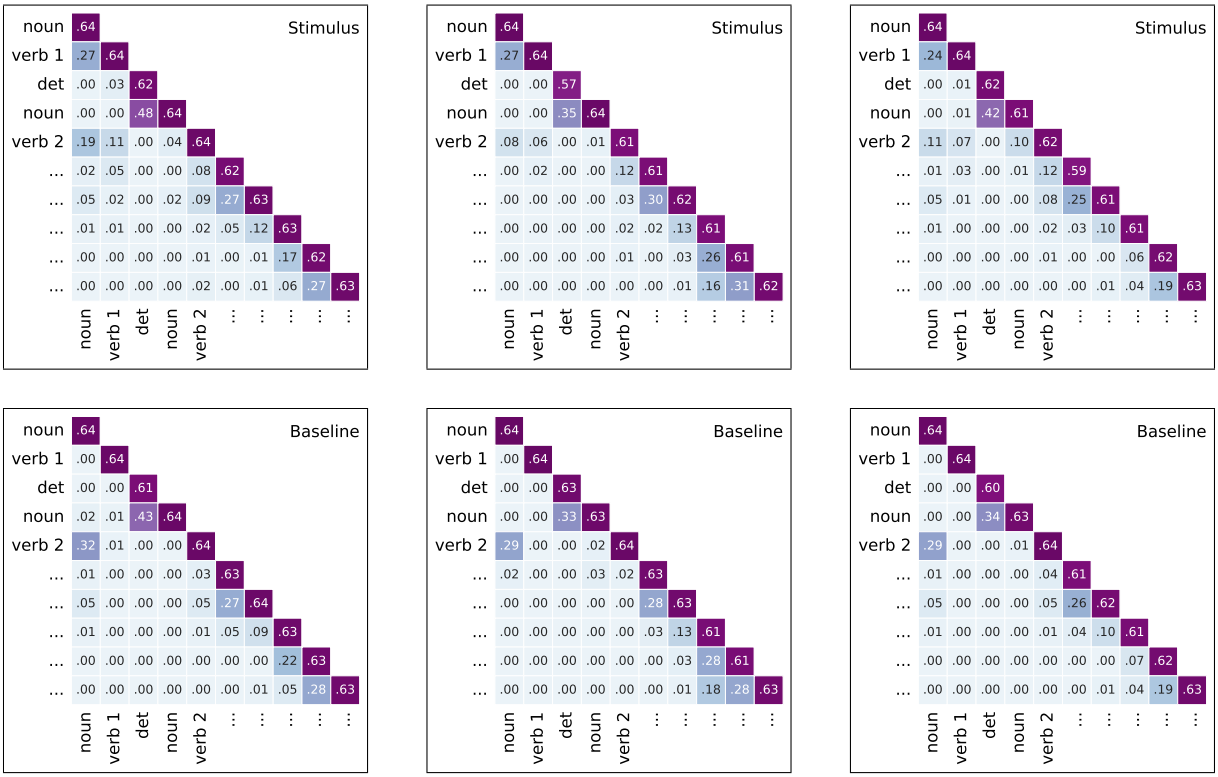


(b) JS divergence wrt. the previous time step

Figure 24: Overview of the JSD for the label attention distributions wrt. the last and previous time steps as reference, averaged over all NP/S stimuli and baselines. From left to right: biaffine parser, DiaParser (ELECTRA-EWT and ELECTRA-PTB).



(a) JS divergence wrt. the last time step



(b) JS divergence wrt. the previous time step

Figure 25: Overview of the JSD for the label attention distributions wrt. the last and previous time steps as reference, averaged over all MVRR stimuli and baselines. From left to right: biaffine parser, DiaParser (ELECTRA-EWT and ELECTRA-PTB).