Communicating in Emergent Language with an Induced Morphological Phrasebook

Anonymous ACL submission

Abstract

We build rule-based emergent language (EL) agents using induced morphological phrasebooks and test their communicative performance in the EL environment with its neural network agents. This contributes three things: First, it assesses the quality of the morphemes discovered by the induction algorithm *in situ*, which we find to be effective for communicating in the EL. Second, it allows us to uncover morphosyntactic properties of EL through ablating the morpheme induction and the phrasebook algorithms, showing that the ELs rely on repetition as well as morpheme ordering to convey meaning. Third, we find that the bijectivity of morphemes (measured via normalized pointwise mutual information), serves as a metric of compositionality that is more closely correlated with the ability of the phrasebook-agents to "speak" and "hear" an EL than existing metrics such as topographic similarity or bag-of-symbols disentanglement.

1 Introduction

Deep learning-based emergent language (a.k.a. emergent communication) presents a fascinating way to study the origins and development of human language by observing how the interplay of functional pressures and inductive biases produce communication systems. Yet a major challenge in studying emergent languages is interpreting how they convey meaning—neural networks may invent communication systems which lack features of human language. Thus, a primary goal of emergent language research has been to investigate the linguistic structures present in emergent languages from morphology to semantics to syntax and how they compare with those of human language (van der Wal et al., 2020; Ueda et al., 2022; Boldt and Mortensen, 2024).

One technique introduced to discover the morphology of emergent language is CSAR (Boldt and Mortensen, 2025), an algorithm which induces morphemes (minimal form–meaning pairs) from an emergent language corpus of messages and their accompanying semantics. CSAR was initially validated on human and procedurally generated languages, yet its effectiveness on emergent languages was not tested due the unavailability of "ground truth" morphemes in emergent languages. To address this shortcoming, we test the morphemes induced by CSAR *in situ* by constructing algorithms to "speak" and "hear" emergent languages using the induced morpheme "phrasebooks" (illustrated in Fig. 1). These phrasebook-based agents are built on the assumption that the emergent language is *concatenatively compositional*, that is, that we can compose the meanings of morphemes by concatenating their forms into a message.

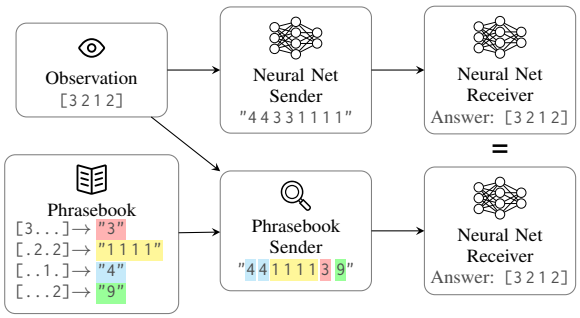


Figure 1: Main experimental setting where we compare a phrasebook-based agent with the original neural network-based sender. We also experiment with a phrasebook-based receiver.

Analyzing the effectiveness of these phrasebook-based agents communicating in emergent language provides two insights. First, it reveals whether or not CSAR is uncovering a meaningful mapping between form and meaning for the emergent language, and, as a consequence, can serve as a good foundation for subsequent study of the structure of emergent languages. Second, it serves a probe into the morphosyntactic properties of emergent

language (when we vary the morpheme induction and phrasebook algorithms).

Finally, with these results, we propose a new compositionality metric: *morpheme bijectivity*, defined as the weighted average of morphemes' normalized pointwise mutual information as induced by CSAR. This metric correlates better with the ability of the phrasebook-based agents (which presuppose compositionality) to use emergent language than existing metrics of compositionality like topographic similarity or bag-of-symbols disentanglement.

Impact Beyond validating the performance of CSAR and characterizing the morphosyntax of a particular set of emergent languages, this paper provides a more general paradigm for more principled evaluation of the linguistic features of emergent languages. That is, by constructing a procedural method for speaking and hearing the emergent language, we can directly probe for its various morphological and syntactical properties. This is much more akin to the process in linguistics of proposing theoretical models and testing their predictions than prior approaches in emergent language. As a whole, this paper, then, moves emergent language research closer towards its goal of discovering the nature of human language.

Structure Section 2 discusses prior work most relevant to this paper. Section 3 presents an overview of the environments, agents, and algorithms uses in the experiments. Section 4 briefly specifies the experiments and their results with a robust discussion in Section 5. The conclusion, discussion of limitations, and ethical considerations are presented in Sections 6 to 8, respectively.

2 Related Work

For a general overview of deep learning-based emergent language see Lazaridou and Baroni (2020). Generally speaking, this work builds on top of the morphological investigations into the structure of emergent language presented in Boldt and Mortensen (2025) and ties in with other approaches, including Ueda et al. (2023); Lipinski et al. (2024); Carmeli et al. (2024); Gilberti et al. (2025).

Levy et al. (2025) addresses the task machine translation of emergent communication into human language; namely, they employ unsupervised neural machine translation to translate image captions between emergent and human language. This is analogous to how we translate between emergent language messages and semantics, although we focus on white-box algorithms instead of deep learning methods. The morpheme induction and phrasebook algorithms assume compositionality in the emergent languages, and as such, intersect with prior work discussing the link between compositionality and generalization in emergent language (Chaabouni et al., 2020; Kharitonov and Baroni, 2020; Xu et al., 2022). Galke et al. (2024) discusses the relationship between compositionality in emergent language and its learnability by deep learning techniques analogous to our study of the learnability by statistical and rule-based methods.

The phrasebook-based agents we propose are, in effect, a rule-based machine translation (RBMT) system (e.g., Forcada et al. (2011)) where the language pair is the message and observation spaces of an emergent language, although the comparative simplicity of emergent language and its divergence from human language admit limited applicability of prior approaches from RBMT techniques with human language.

3 Methods

The code for this paper available under a free license at `http://example.com/supplemental-materials-for-review`.

3.1 Environment

The emergent language environment we use for our investigation is the reconstruction variant of the signalling game (similar to Chaabouni et al. (2019, 2020)). The reconstruction game comprises two agents, a sender and a receiver. The sender makes an observation and produces a message which the receiver must use to reproduce the original observation (without any additional input); this can be seen as mimicking an autoencoder architecture where the central bottleneck layer is a sequence of discrete symbols. We select the reconstruction game because it in addition to being well-studied in literature, it provides a clear way of testing generalization via a held-out test set. Our implementation is based on the EGG framework (Kharitonov et al., 2021, MIT license).

In our implementation, the observations are a concatenation of one-hot vectors, which each represent a distinct attribute taking on a particular value. For example, if we have a game where observations

comprise 2 attributes which can each take on one of 3 values, we would resent the observation $[0\ 2]$ as follows:

$$[0\ 2] \underset{\text{one-hot}}{\rightarrow} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \underset{\text{flatten}}{\rightarrow} [1\ 0\ 0\ 0\ 0\ 1],$$

with the concatenated one-hot vector being the input to the sender and desired output from the receiver. The messages in our environment are represented similarly: sequences of tokens represented as one-hot vectors (although the one-hot vectors are not concatenated because they are processed sequentially). Observations are sampled uniformly from all of the possible combinations of values for each attribute. Agents only see a subset of these possible observations at training time with the rest being held out for a test set.

3.2 Morpheme induction

Morpheme induction is the process of inferring minimal, meaningful form–meaning pairs from a corpus of messages with their accompanying meanings (i.e., an *annotated* corpus). Messages are taken to be sequences of atomic tokens while the accompanying meanings are represented as sets of atomic meaning tokens. Forms, in this case, are subsequences of tokens and meanings are subsets of the observation made by the sender. A given form–meaning pair is *meaningful* if the form and meaning correspond to each other in the language and *minimal* if the pair cannot be further decomposed into simpler pairs while maintaining the same meaning.

We use the CSAR algorithm to induce morphemes from the annotated emergent language corpora produced by the emergent language game (Boldt and Mortensen, 2025, MIT license). Its outline is:

C ount co-occurrences of forms and meanings.
S elect pair with highest mutual information.
A blate selected pair from corpus.
R epeat process, starting at *Count*.

The intuition is that form–meaning pairs with high mutual information both correlate well with each other as well as occur frequently within the corpus.

3.3 Agents

Neural networks The sender agent comprises two fully-connected embedding layers, separated by a tanh activation, mapping from the observation to the embedding space which is given to an RNN

Algorithm 1 Outline of sender algorithm

```
morpheme :: tuple
  form :: list[int]
  meaning :: set[int]
  weight :: float
# Sorted by weight, descending
inventory :: list[morpheme]
observation :: set[int]
max_length :: int

message :: list[int]
while True:
  meanings_left = observation
  for morph in inventory:
    if morph.meaning ⊆ meanings_left:
      meanings_left -= morph.meaning
      message.extend(morph.form)
      if len(message) ≥ max_length:
        return message
      if meanings_left is empty:
        break # out of for loop
```

to produce the message. The discrete vocabulary items are sampled using a Gumbel-Softmax layer (Maddison et al., 2017; Jang et al., 2017) before being passed to the RNN of the receiver agent, follows by two full-connected layers, as in the sender. The agents are trained on a mean-squared error objective. When the exact-match training accuracy reaches 99.99%, either the sender or receiver's parameters are reset to encourage better generalization (by default the receiver is reset twice before concluding training after reaching the accuracy threshold) (Li and Bowling, 2019).

We use two different types of neural network-based agents in our empirical evaluation which we term *online* and *offline*. The online agents are simply the neural networks that were trained as part of the original emergent language environment, which means that the sender and receiver were jointly optimized and gradients backpropagated from receiver to sender. The offline agents, on the other hand, are trained directly from the annotated corpora in a fully supervised setting. Thus, the offline agents do not receive any additional information or benefit from joint optimization compared to the phrasebook-based agents which also operate based only on the annotated corpora.

Phrasebook The phrasebook agents map from observations to messages (senders) or messages to observations (receivers) by executing an algorithm given the morpheme inventory ("phrasebook") induced by CSAR. Both algorithms are founded on the assumption that the emergent languages are *concatenatively compositional* over morphemes;

Algorithm 2 Outline of receiver algorithm

```
# Sorted by weight, descending
inventory :: list[morpheme]
message :: list[int]
attributes :: list[set[int]]

# Cumulative weight of each meaning
meanings :: map[int, float]
for morph in inventory:
  n = count_matches(message, morph.form)
  for m in morph.meaning:
    meanings[m] += n * morph.weight
# Max-weighted value for each attribute
observation :: set[int]
for attr in attributes:
  observation += argmax(meanings ∩ attr)
return observation
```

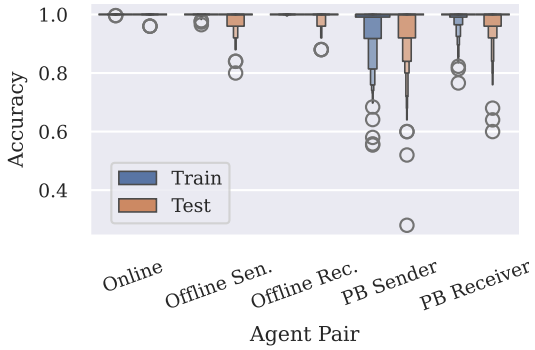


Figure 2: Accuracies across ∼1200 random seeds for various agent pairs. *Online* and *Offline* are the neural network-based agents, while *PB* refers to the phrasebook-based agent.

that is, the meanings present in morphemes can be combined by concatenating their corresponding forms.

The sender uses a greedy algorithm (Algorithm 1) which loops over the morphemes in the phrasebook from best to worst; when a morpheme's meaning is a subset of the observation, the corresponding form is added to the message under construction. The morpheme's meaning is them ablated from the observation, and the loop continues until all the meanings are matched or the message reaches the maximum length permitted by the environment. If the observation is exhausted before the max length is reached, the observation is reset to the original value and algorithm repeats from the beginning of the morpheme inventory. This results in multiple of the highest-weighed morphemes being employed in the message (this turns out to be critical to phrasebook sender performance; see Section 5.3). When the maximum message length is reached, the morphemes are reordered by ascending positional *affinity* before being concatenated. Positional affinity is defined as the mean start index of that morpheme across the input corpus.

We also implement more sophisticated algorithms for constructing messages from an observation: best-first search and integer programming (used in the ablation experiments and briefly described in Appendix B).

The receiver algorithm (Algorithm 2) follows the same basic idea as the sender. The morphemes are looped over, testing to see if a morpheme's form is a subsequence of the message. If it the form is contained in the message, the morpheme's meaning is added to a meaning accumulator proportional to the morpheme's weight and the number of occurrences in the message. Unlike the sender, the form is not subtracted from the message, and the algorithm proceeds through the whole phrasebook only one time; thus, a sort of superposition of all mathcing morphemes is considered. After the phrasebook is exhausted, the final meaning is determined by taking the index of the max accumulator value for each attribute.

4 Experiments

In this section, we describe the experiments performed and give their results with a full discussion and analysis of the results in Section 5. See Appendix C for details on computing resources used.

4.1 Phrasebook vs. neural network agents

We begin our empirical evaluation by comparing the performance of the phrasebook-based agents with the online and offline neural network-based agents across ∼1200 seeds of the reconstruction game (Fig. 2). We look at exact-match accuracy on train and test sets for the original online–online setting as well as to and from the offline and phrasebook agents. This setting uses 4-attribute, 4-value observations which means that random chance performance is $\frac{1}{4^4} \approx 0.4\%$ (see Appendix A for further hyperparameters). Some qualitative examples of inventories and messages sent by the neural sender are given in Appendix D. A quantitative summaries of the morpheme inventories (e.g., inventory size, synonymy) is included in Appendix E.

4.2 Environment ablation

We follow up our primary evaluation of our phrasebook-based agents with variations of the hyperparameters of the emergent language environ-

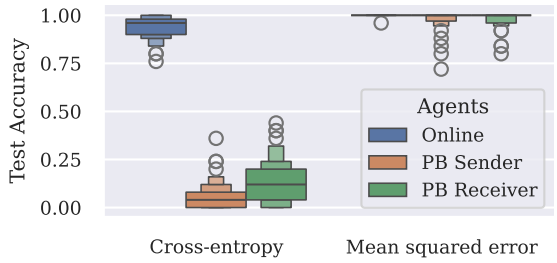


Figure 3: Accuracy plot for loss function ablation experiment. Full plots in Appendix F.

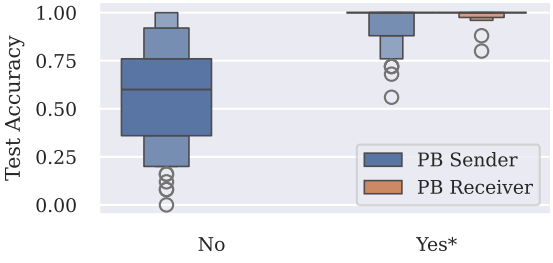


Figure 4: Accuracy plots various for *repeat morpheme* phrasebook sender ablation. Full plots in Appendix G.

ment (and the online agents) to determine what environmental factors make emergent languages more or less intelligible to morpheme induction and the accompanying phrasebook agents. An example of the results is shown in Fig. 3 with all results given in Appendix F. Each setting was run for 100 random seeds (some settings had fewer than 100 runs converge in the initial training).

In particular, we vary the following hyperparameters in the environment and online agents:

***n* attributes, *n* values** (default: $(4, 4)$).

test proportion Proportion of possible observations held out for the test set (default: 10%).

vocabulary size Number of distinct types usable in a message (default: 64).

max message length Maximum number of tokens in a message; padded with end-of-sentence token if less than max (default: 8).

GS straight through Use a stochastic one-hot vector during the forward pass through the Gumbel-Softmax layer (default: no).

Agent reset schedule Sequence of sender/receiver resets during training (default: (receiver, receiver)).

Reset optimizer parameters Reset Adam optimizer parameters along with the agent parameters during reset (default: yes).

RNN cell type Which RNN cell type to use (Elman, LSTM, GRU; default: GRU).

loss function Which loss function to train the online agents with (cross-entropy or mean squared error; default: MSE).

***n* embedding layers** Number of fully-connected layers before/after the sender's/receiver's RNN (default: 2).

***n* RNN layers** (default: 1)

4.3 Phrasebook agent ablation

In this section we test ablations and variations of the morpheme induction process as well as the phrasebook sender and receiver algorithms. These variations will elucidate what aspects of the algorithms and morphemes are most salient to using the emergent language. An example of the results is shown in Fig. 4 with all results given in Appendix G.

In the morpheme induction algorithm we vary:

max inventory size Use the top-k morphemes for the inventory (default: $k = \infty$).

max form length Consider forms up to length l (default: $l = \infty$).

max meaning size Consider meanings up to size s (default: $s = \infty$).

weight method What probabilistic metric to use in selecting weighting and selecting morpheme candidates (default: mutual information); other methods given in Appendix G.

strip EoS token Remove the padding/end-of-sentence token from messages before induction (default: yes).

search best Apply a lookahead heuristic in ablating ambiguous morphemes (default: yes).

In the phrasebook sender we vary:

morpheme selection method Whether to use greedy algorithm (default) for selecting morphemes, search, or integer programming.

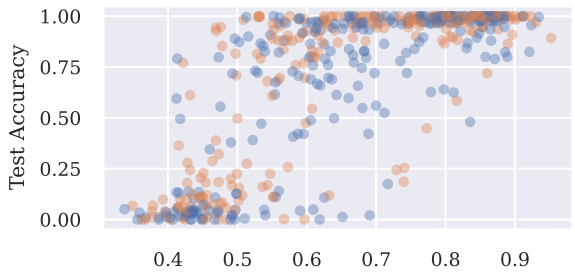
form order How to order the forms in the message (default: *affinity*); other method includes *insertion* order (no reordering) and *shuffled* (at the morpheme level).

repeat morpheme Permit repetitions of the same morpheme (default: yes); if no, exit after all meanings are accounted for.

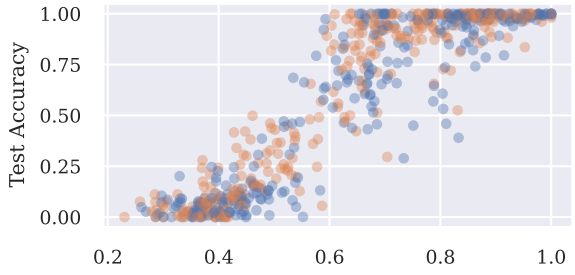
ablate meaning Track meaning atoms remaining in the observations (default: yes); if no, perform a single pass through the inventory and apply any matching morpheme (increases diversity of morphemes used).

In the phrasebook receiver we vary:

ablate form Mask out form tokens as they are



(a) Corpus topographic similarity ($R^2 = 0.57$)



(b) Morpheme bijectivity ($R^2 = 0.85$)

Figure 5: Plot of two different potential predictors of phrasebook sender (blue) and receiver (orange) accuracy: corpus topographic similarity (top) and mean NPMI of morphemes (bottom).

matched with morphemes (default: no); if no, match all morphemes in the inventory that apply.

idempotent form Ignore repetition of the same morpheme in a message (default: no); analogous to *repeat morpheme* for the sender algorithm.

Each variation is applied to the base configuration independently and is tested against the same 100 corpora generated with different random seeds.

4.4 Compositionality

In our final experiment, we investigate how the performance of the phrasebook agents correlates with topographic similarity and a compositionality metric we introduce, *morpheme bijectivity* (Fig. 5).We also tested bag-of-symbols disentanglement (Chaabouni et al., 2020) and a mutual information-based variant of the morpheme bijectivity metric (Fig. 12). These metrics are computed reusing the training corpora and phrasebook agent accuracies from the environment ablation experiments (Section 4.2) in order to test against a wider variety of languages and phrasebook agent performances.[1] Since the independent variable values (i.e., toposim and intrinsic inventory metrics) are not evenly distributed, we employ stratified sampling across 5 buckets to avoid overweighting the most prevalent buckets.

Topographic similarity (a.k.a., "toposim") (Brighton and Kirby, 2006; Lazaridou et al., 2018) is the most popular metric of compositionality in the emergent language literature and is defined as the Spearman rank correlation coefficient (ρ) between pairwise distances in the observation space and distances in the message space. We use Levenshtein distance as the metric for the message space and Hamming distance for the observation space (since the observations are represented by sets of a fixed size).

We compare toposim with a new metric, *morpheme bijectivity*, derived from CSAR's morpheme inventory. Morpheme bijectivity is defined as the mean normalized pointwise mutual information[2] (NPMI) weighted by the prevalence[3] of morphemes in the inventory given by CSAR. The intuition with mean NPMI is that morphemes with higher NPMI's (i.e., closer to having a one-to-one relationship) result in observation–message mappings that are less ambiguous and easier to process.

5 Discussion

5.1 Communicating in emergent language

We observe (in Fig. 2) that the online–online parings perform near 100% accuracy on training and test data with few outliers. Both the offline sender and offline receiver perform almost as well, though with a marked decrease on the test data; this shows that most—though possibly not all—of in the information necessary for generalizing to the test data is is available in the annotated corpora alone (cf. the online agents "overfitting" to each other).

The phrasebook-based agents, while underperforming the neural network-based agents, perform quite well with median scores of 100% on the test set and seldom falling below 80%. This, in turn, confirms the primary premise in question in this paper: it is possible to communicate in emergent language with phrasebook agents derived from an induced morphological inventory. Nevertheless, it is very easy to distinguish between the messages generated by the neural sender compared to the phrasebook sender (Appendix D); the former is more varied in both its morpheme selection and its

[1] We exclude variations on the hyperparameters n values, n attributes, and test proportion because they skew the relative difficulty of the reconstruction game.

[2] Computed for form–meaning pairs before any ablations.

[3] The joint probability of the form–meaning pair after previous ablations are applied.

order compared to the rigid message structure of the phrasebook sender.

5.2 Robustness of morpheme induction and phrasebook agents

Varying the hyperparameters of the emergent language environment and the online agents allows us to get a fuller picture of how the morpheme induction, sender, and receiver algorithms perform (Figs. 3, 8 and 9). We categorize the results by the online and phrasebook agent performances:

High online, high phrasebook Both the online agents and the phrasebook agents perform well ($>80\%$)[4], suggesting that generalization is achieved by the online agents employing a method interpretable to both sender and receiver phrasebook agents. This includes the majority of the settings we tested in our ablations suggesting a fair degree of robustness to variations in the hyperparameters of the environment.

Low online, low phrasebook Both the online agent and the phrasebook agents struggle to generalize well (even after the online agents converge on the training set). These settings include $n_{\text{attribute}} = 2$ (0%) and $n_{\text{value}} = 2$ (0%) and, to a lesser degree, a test proportion of 75% ($[30\%, 45\%]$). The results on these settings are expected insofar as the online agents serve as an upper bound for phrasebook agent performance, that is, the phrasebook agents cannot somehow generalize beyond what possible in the original language.

High online, low phrasebook The online agents maintain a high test performance while the phrasebook agents significantly underperform the online agents; the most notable examples of this are the non-GRU RNN cell type ($[90\%, 100\%]$ vs. $[15\%, 35\%]$), cross-entropy loss (95% vs. $[5\%, 15\%]$), and a vocabulary size of 4 (100% vs. 5%). These setting highlight the fact that the online agents are capable of generalizing to unseen observation in a way that is largely opaque to the morpheme induction algorithm and/or the phrasebook algorithms, that is, in a way that violates the assumption concatenative compositionality. For example, in the case of a small vocabulary, size it would not be possible to create a one-to-one mapping between specific types in the vocabulary and specific values for specific attributes. Two alternatives for conveying meaning include: (1) double articulation where groups of meaningless form tokens comprise meaningful "words" and (2) the semantics of individual types exist in an entangled, non-linear embedding space which neural networks can learn but CSAR cannot. Although further investigation would be required to determine if either of these two cases hold.

[4] Parenthesized numbers are median accuracies.

5.3 Morphosyntax in the ELs

Looking at the performance variations in response to changing parameters of the induction and phrasebook agents, we can infer various properties about the morphology and syntax of the emergent languages in question. With respect to the morpheme induction, we see two subtle patterns: (1) performance goes slightly up when increasing the max meaning size from 1 to 2 (Fig. 10) and (2) performance goes slightly *down* with max form length increasing above 1 (Fig. 10). The former pattern suggests that the emergent languages show some small degree of fusionality/cumulative exponence, while the latter indicates that considering multi-token forms is not relevant to the morphology of the emergent languages.

Turning our attention to the variations in the sender algorithm, we see that one of the most important factors in high performance is permitting the repetition of morphemes. When the algorithm employs just a single morpheme per meaning atom, the performance is notably worse (with a median accuracy of 60% instead of 100%; Fig. 11). Similarly, when the sender algorithm does not ablate meanings and uses subsequent lower-weighed morphemes for a given meaning atom in addition to the higher weighed meanings (i.e., meanings are not ablated), performance drops (Fig. 11). These observations suggest that repetition of highly-weighted morphemes is a meaningful morphosyntactic feature of the emergent languages studied.

In addition to the importance of repetition, we also found the order of morphemes had a notable impact on the sender agent's performance, with performance decreasing when we perturb morphemes from their preferred place in the message (i.e., the *affinity* method; Fig. 11). This indicates a syntactic rule (albeit a simple one) that governs how morphemes should be combined. Finally, the variations on the receiver algorithm did not have large impacts on the phrasebook agent performance (Fig. 11).

5.4 Compositionality in emergent language

Compositionality is one of the most talked about topics in emergent language research, and the *in situ* experiments with phrasebook agents provide a uniquely in-depth investigation of this phenomenon. Namely, the morpheme induction algorithm and phrasebook agents are built on the following assumption: the meanings forming the observation correspond to substrings of the message in a disentangled way, such that the substrings can otherwise be recombined independently of each other while retaining their original meaning. This aligns closely with the typical definition of compositionality in emergent language, namely that "[t]he meaning of a compound expression is a function of its parts and of the way they are syntactically combine" (Partee et al., 1984), where syntactic combination is operationalized as concatenation of the forms in emergent language.

Essentially, the phrasebook agents should perform well if and only if (1) the emergent language is concatenatively compositional, (2) the morpheme induction algorithm actually induces meaningful morphemes, and (3) the sender and receiver algorithms sufficiently capture the syntax of the emergent language. Insofar as we assume (2) and (3) to be true, we can use the performance of the phrasebook agents as a proxy for compositionality. This notion of compositionality is deeper than more familiar metrics like topographic similarity because it not only takes into account the externally visible correlations between observations and messages but also tests these correlations with the original emergent language speakers and hearers (i.e., the online agents). We could, in a way, see these experiments being closer to "field linguistics" compared to the "corpus linguistics" approach of toposim. Continuing the analogy: field linguistics is more resource intensive than corpus linguistics, and likewise for testing phrasebook agents *in situ* compared to computing toposim.

Yet, we find that morpheme bijectivity addresses the shortcomings of both measures of compositionality: First, it is more predictive of phrasebook agent performance than toposim (Fig. 5; $R^2 = 0.86$ vs. $R^2 = 0.57$).[5] And second, it can be computed with only the annotated corpus, not needing access to the original emergent language environment.

[5] Bag-of-symbols underperforms both of these as a predictor with $R^2 = 0.29$ (Fig. 12).

Compositionality without generalization An ancillary finding of the experiments across different environments is that they provide further evidence for the notion that neural network agents do not need to use concatenative compositionality to in order to generalize (Kharitonov and Baroni, 2020). Indeed, in the majority of our settings, we see that the performance of the online and phrasebook agents are relatively matched, suggesting that compositionality corresponds with generalization performance. But in certain settings (e.g., cross-entropy loss, Elman and LSTM RNN cells, and a small vocabulary size of 4; Figs. 8 and 9), there is a clear bifurcation: the online agents generalize well while the phrasebook agents do not (and even have trouble fitting the training data). This shows that the neural network agents are capable of compositional as well as non-compositional generalization and may favor one over the other for non-obvious reasons (cf. GRUs vs. LSTMs). Furthermore, the lower correlation of toposim with phrasebook agent performance suggests that it might not be the best metric for detecting such instances of non-compositionality in comparison to the higher predictive power of morpheme bijectivity derived from CSAR.

6 Conclusion

The above experiments have demonstrated the effectiveness of CSAR and the phrasebook algorithms as foundations for studying the morphological and syntactic structures of emergent language. Beyond the particular languages studied here, the methods presented demonstrate a more general paradigm of investigating the linguistic features of emergent languages by ablating explicit models of communication and testing them *in situ*. Such an approach is better grounded in the use of the emergent language itself rather than merely hypothesizing about the relationship between surface-level features and their relationship with the language itself. With the tools and techniques presented in this paper, the field of emergent language is better equipped to explore the nature of human language.

7 Limitations

We identify the following primary limitations in our work:

1. The phrasebook sender and receiver algorithms were designed more or less heuristically rather than being motivated by some par-

ticular feature of how the online agents convey meaning in emergent language. A more principled design approach, grounded in how the neural network agents express meaning, could yield even better performance for the phrase-book agents.

2. We have performed limited qualitative of evaluation of the emergent languages, and in particular the strategies used by the online neural network agents to generalize without simple compositionality. Observing the emergent languages more closely is an important step into figuring out precisely why morpheme induction and/or phrasebook algorithms fail while the neural networks succeed.
3. Although the methods we present are applicable to a wide variety of emergent language environments, the empirical evaluations were performed with a limited variety of environments, so it is not possible to determine whether or not the experimental results are fully applicable to other environments from the presented experiments alone.
4. The morpheme bijectivity metric is limited to the emergent languages that CSAR can handle, specifically emergent languages with discrete, decomposable observations. While CSAR, and thereby morpheme bijectivity, could theoretically be expanded, it currently cannot be applied to continuous observations.

8 Ethical Considerations

We do not identify any ethical considerations or potential risks relevant to this work as it constitutes basic machine learning and linguistics research using synthetic data.

References

Brendon Boldt and David R Mortensen. 2024. A review of the applications of deep learning-based emergent communication. *Transactions on Machine Learning Research*.

Brendon Boldt and David R. Mortensen. 2025. Morpheme induction for emergent language. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25275–25290, Suzhou, China. Association for Computational Linguistics.

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial Life*, 12(2):229–242.

Boaz Carmeli, Yonatan Belinkov, and Ron Meir. 2024. Concept-best-matching: Evaluating compositionality in emergent communication. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3186–3194, Bangkok, Thailand. Association for Computational Linguistics.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online. Association for Computational Linguistics.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019. Anti-efficient encoding in emergent communication. *arXiv*, 1905.12561.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Lukas Galke, Yoav Ram, and Limor Raviv. 2024. Deep neural networks and humans both benefit from compositional language structure. *Nature Communications*, 15(1).

Miles Gilberti, Shane Storks, and Huteng Dai. 2025. Discovering properties of inflectional morphology in neural emergent communication. *arXiv*, 2508.05843.

Eric Jang, Shixian Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 2017 International Conference on Learning Representations (ICLR)*.

Eugene Kharitonov and Marco Baroni. 2020. Emergent language generalization and acquisition speed are not tied to compositionality. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 11–15, Online. Association for Computational Linguistics.

Eugene Kharitonov, Roberto Dessì, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2021. EGG: a toolkit for research on Emergence of lanGuage in Games. `https://github.com/facebookresearch/EGG`.

Angeliki Lazaridou and Marco Baroni. 2020. Emergent multi-agent communication in the deep learning era. *arXiv*, 2006.02419.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv*, abs/1804.03984.

Ido Levy, Orr Paradise, Boaz Carmeli, Ron Meir, Shafi Goldwasser, and Yonatan Belinkov. 2025. Unsupervised translation of emergent communication. *arXiv*, 2502.07552.

Fushan Li and Michael Bowling. 2019. Ease-of-teaching and language structure from emergent communication. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Olaf Lipinski, Adam Sobey, Federico Cerutti, and Timothy J. Norman. 2024. Speaking your language: Spatial relationships in interpretable emergent communication. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 2017 International Conference on Learning Representations (ICLR)*.

Barbara Partee and 1 others. 1984. Compositionality. *Varieties of formal semantics*, 3:281–311.

Ryo Ueda, Taiga Ishii, and Yusuke Miyao. 2023. On the word boundaries of emergent languages based on harris's articulation scheme. In *The Eleventh International Conference on Learning Representations*.

Ryo Ueda, Taiga Ishii, Koki Washio, and Yusuke Miyao. 2022. Categorial grammar induction as a compositionality measure for emergent languages in signaling games. In *Emergent Communication Workshop at ICLR 2022*.

Oskar van der Wal, Silvan de Boer, Elia Bruni, and Dieuwke Hupkes. 2020. The grammar of emergent languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3339–3359, Online. Association for Computational Linguistics.

Zhenlin Xu, Marc Niethammer, and Colin Raffel. 2022. Compositional generalization in unsupervised compositional representation learning: A study on disentanglement and emergent language. *arXiv*, 2210.00482.

A Environment Details

The following hyperparameters (for the main experiment) were selected to optimize for online and phrasebook agent performance base, in part, on the environment and agent ablation experiments. The neural network architecture of the online agents is as follows:

- Observation (concatenated one-hot vectors)
- Sender
  - Feed-forward layer (embedding layer)
  - Tanh activation layer
  - Feed forward layer
  - RNN (w/ Gumbel-Softmax layer)
- Message (max length $+1$ by vocabulary size matrix; last token is an end-of-sentence token.)
- Receiver
  - RNN
  - Feed-forward layer (embedding layer)
  - Tanh activation layer
  - Feed forward layer
  - Projection layer to observation space
- Prediction (same dimension as observation)

The offline agents have the same architecture. The offline sender is trained with cross-entropy loss after the bottleneck layer. The offline receiver is trained with mean squared error after the projection layer (like the online receiver).

Our environment uses the following hyperparameters.

accuracy threshold 0.9999; training accuracy value at which next agent reset is performed or training is stopped if all resets have been performed.

neural network resets (receiver, receiver); sequence of resets of sender/receivers after accuracy threshold is reached.

max epochs 300; epochs after which training is stopped and the trial is considered a failure.

observation distribution uniform; distribution from which the values for the attributes are drawn from.

***n* attributes** 4; number of attributes in the observation.

***n* values** 4; number of distinct values in that each attribute can take.

test proportion 10%; proportion of unique observations that are held out from training for the test set to gauge generalization performance.

batch size 2^{10}; batch size of the neural network agents during training.

loss function mean squared error; objective function used for backpropagation during training; the other loss function implemented and tested is cross-entropy loss.

learning rate 1.8×10^{-3}; learning rate for neural network agents.

learning rate schedule none; learning rate was left constant during training.

***n* examples per epoch** 2^{14}; number of examples shown to the neural network agents before checking the accuracy and possibly performing rests or concluding training.

optimizer AdamW; optimizer for the neural network agents.

weight decay 1×10^{-10}; weight decay applied through the optimizer.

reset optimizer yes; whether or not to reset the optimizer's parameters for the agent that is reset as determined by the reset schedule.

max message length 8; maximum length of the message from the sender agent; the final message passed to the receiver is always one token longer than the maximum length as the message from the sender is padded with one or more end-of-sentence/pad tokens (the same token).

discrete optimization method Gumbel-Softmax; the method used for handling the discrete nature of the senders methods; REINFORCE is another method but was not used in the experiments.

sender RNN cell type GRU; other options include Elman ("RNN") and LSTM.

receiver RNN cell type *as above*.

sender RNN layers 1.

receiver RNN layers 1.

sender hidden size 256; number of hidden units in the sender's RNN.

receiver hidden size *as above, mutatis mutandis*.

sender embedding size 128; number of units in the embedding (feed-forward) layers in the sender.

receiver embedding size *as above, mutatis mutandis*.

sender *n* embedding layers 2; number of embedding layers before the RNN.

receiver *n* embedding layers 2; number of embedding layers after the RNN.

Gumbel-Softmax straight through no; whether or not to use sampled one-hot values for the vectors during the forward pass while using the categorical values during the backward pass.

Gumbel-Softmax temperature 1.

vocabulary size 64; number of possible distinct types in the message, that is, the size of the one-hot vectors forming the message; this number includes the reserved end-of-sentence/pad token.

B Alternative Sender Algorithms

Both implementations of more sophisticated algorithms for the phrasebook-based sender resulted inferior performance. We speculate that the primary reason for this is that the greedy approach of selecting the highest weighted morphemes (which match the desired meaning) and repeating them as much as would fit matches the strategy of conveying meaning more than the inductive biases of the more sophisticated approaches.

Search This algorithm implemented best-first search where the objective was to account for each component of the observation by selecting a morpheme while minimizing the sum of morpheme costs, defined as the negative weight of the morpheme assigned by CSAR.

Integer Programming In this sender algorithm the meaning of each morpheme is treated as binary vector scaled by the weight given by CSAR for that morpheme. An integer programming solver would then try to maximize the distance per attribute between the correct value and the highest incorrect value by selecting a non-negative number of occurrences for each morpheme while staying within the max length.

C Computing Resources

The environment employed for this paper requires a GPU with modest memory requirements ($<$1 GiB). Altogether, the experiments used $\sim$200 GPU-hours on an Nvidia L40S or equivalent.

D Qualitative Results

Example inventories given in Tables 1 to 3. Example of messages generated by the neural and phrasebook agents (derived from Table 1) given below:

Example 1 Neural sender: "59 59 22 22 2 2 61 61 0" interpreted as: 3133. Phrasebook sender: "59 59 59 59 22 2 61 61 0" interpreted as: 3133. Used the following morphemes: ("59 59", .1..) ("2", ...3) ("22", ..3.) ("61 61", 3...) ("59 59", .1..)

Example 2 Neural sender: "25 25 46 62 2 46 2 62 0" interpreted as: 0213. Phrasebook sender: "25 25 46 46 62 62 2 2 0" interpreted as: 0213. Used the following morphemes: ("25", .2..) ("46", ..1.) ("62", 0...) ("2", ...3) ("25", .2..) ("46", ..1.) ("62", 0...) ("2", ...3)

Example 3 Neural sender: "59 59 62 31 38 62 38 31 0" interpreted as: 0122. Phrasebook sender: "59 59 59 59 31 31 62 38 0" interpreted as: 0122.

Form	Meaning	Weight	Proportion
22	2...	8.20 e−1	2.55 e−1
29	.1..	8.20 e−1	2.55 e−1
47	...1	8.20 e−1	2.55 e−1
28	.3..	8.13 e−1	2.51 e−1
40	..0.	8.13 e−1	2.51 e−1
60	0...	8.13 e−1	2.51 e−1
48	...2	7.99 e−1	2.42 e−1
7	.0..	7.99 e−1	2.42 e−1
8	..2.	7.99 e−1	2.42 e−1
9	..1.	7.99 e−1	2.42 e−1
61	3...	7.77 e−1	2.29 e−1
3	1...	7.65 e−1	2.55 e−1
54	...0	7.45 e−1	2.42 e−1
27	..3.	5.08 e−1	1.99 e−1
15	...3	5.02 e−1	1.90 e−1
43	.2..	5.02 e−1	1.90 e−1
2	.2.3	3.30 e−1	6.06 e−2
34	..3.	1.34 e−1	6.49 e−2
2	.2..	1.30 e−1	6.06 e−2
2	...3	1.30 e−1	6.06 e−2
33	1..0	3.92 e−2	8.66 e−3
33	...0	3.92 e−2	8.66 e−3
33	1...	3.92 e−2	8.66 e−3

Table 1: A typical morpheme inventory.

Used the following morphemes: ("31", ..2.) ("59 59", .1..) ("62", 0...) ("38", ...2) ("31", ..2.) ("59 59", .1..)

Example 4 Neural sender: "30 30 4 2 4 2 35 35 0" interpreted as: 1003. Phrasebook sender: "30 30 4 4 2 2 35 35 0" interpreted as: 1003. Used the following morphemes: ("30", .0..) ("4", ..0.) ("2", ...3) ("35", 1...) ("30", .0..) ("4", ..0.) ("2", ...3) ("35", 1...)

Example 5 Neural sender: "22 30 22 30 57 19 8 8 0" interpreted as: 2031. Phrasebook sender: "30 30 22 22 57 57 8 8 0" interpreted as: 2031. Used the following morphemes: ("30", .0..) ("8", 2...) ("22", ..3.) ("57", ...1) ("30", .0..) ("8", 2...) ("22", ..3.) ("57", ...1)

Example 6 Neural sender: "22 30 22 30 38 8 38 8 0" interpreted as: 2032. Phrasebook sender: "30 30 22 22 38 38 8 8 0" interpreted as: 2032. Used the following morphemes: ("30", .0..) ("8", 2...) ("22", ..3.) ("38", ...2) ("30", .0..) ("8", 2...) ("22", ..3.) ("38", ...2)

E Quantitative Summary of Results

In Figs. 6 and 7, we illustrate the distributions of various quantitative metrics from the main experiment (Section 4.1) using various metrics defined below. With the exception of *Inventory size*, the following are weighted by the *normalized preva-*

Form	Meaning	Weight	Proportion
34	...2	8.26 e−1	2.60 e−1
7	..0.	8.26 e−1	2.60 e−1
25	..2.	8.20 e−1	2.55 e−1
36	3...	8.13 e−1	2.51 e−1
43	...0	8.06 e−1	2.47 e−1
1	...1	7.99 e−1	2.42 e−1
47	..1.	7.92 e−1	2.38 e−1
16	.1..	7.85 e−1	2.34 e−1
40	1...	5.60 e−1	2.08 e−1
58	..3.	5.46 e−1	1.99 e−1
48	.2..	4.99 e−1	1.99 e−1
53	0...	4.63 e−1	1.77 e−1
17 17	02..	3.63 e−1	6.93 e−2
24 24	1.3.	2.55 e−1	4.76 e−2
54	.0..	2.49 e−1	2.55 e−1
54	...3	2.44 e−1	2.51 e−1
54	2...	2.39 e−1	2.47 e−1
17	0...	1.52 e−1	6.93 e−2
17	.2..	1.42 e−1	6.93 e−2
24	..3.	1.21 e−1	4.76 e−2
24	1...	1.18 e−1	4.76 e−2
34 34	.3..	1.79 e−2	8.66 e−3
54	.3..	8.92 e−3	9.09 e−2
47 34 53	.3..	8.89 e−3	4.33 e−3
43 43	.3..	8.89 e−3	4.33 e−3
40 34	.3..	8.89 e−3	4.33 e−3
53	.3..	7.83 e−3	4.76 e−2
36 43 58	.3..	1.95 e−3	4.33 e−3
43 47 40	.3..	1.95 e−3	4.33 e−3
47	.3..	5.88 e−4	8.66 e−3
25 1	.3..	3.87 e−4	4.33 e−3
36	.3..	3.42 e−4	3.46 e−2
25 34	.3..	3.33 e−4	4.33 e−3
25	.3..	8.07 e−5	4.33 e−3
7 1	.3..	6.62 e−5	4.33 e−3
40	.3..	1.84 e−5	4.33 e−3
34	.3..	1.84 e−5	4.33 e−3
1	.3..	3.40 e−6	4.33 e−3

Table 2: An outlier inventory with very poor performance.

Form	Meaning	Weight	Proportion
4 4	...3	8.33 e−1	2.64 e−1
31	...0	8.26 e−1	2.60 e−1
40 40	3...	8.20 e−1	2.55 e−1
28	2...	8.20 e−1	2.55 e−1
49	1...	8.13 e−1	2.51 e−1
60	...1	8.13 e−1	2.51 e−1
17	0...	7.92 e−1	2.38 e−1
41	...2	7.69 e−1	2.25 e−1
1 1 1	.00.	3.63 e−1	6.93 e−2
26 26 26	.30.	3.63 e−1	6.93 e−2
42 42 42	.10.	3.63 e−1	6.93 e−2
61 61 61	.01.	3.63 e−1	6.93 e−2
15 15 15	.33.	3.47 e−1	6.49 e−2
18 18 18	.31.	3.47 e−1	6.49 e−2
33 33 33	.22.	3.47 e−1	6.49 e−2
43 43 43	.20.	3.47 e−1	6.49 e−2
8 8 8	.11.	3.47 e−1	6.49 e−2
10 10 10	.13.	3.30 e−1	6.06 e−2
13 13 13	.21.	3.30 e−1	6.06 e−2
23 23 23	.32.	3.30 e−1	6.06 e−2
29 29 29	.12.	3.12 e−1	5.63 e−2
38 38 38	.02.	3.12 e−1	5.63 e−2
16 16 16	.03.	2.95 e−1	5.19 e−2
55 55 55	.23.	2.95 e−1	5.19 e−2
61	.0..	8.22 e−2	3.90 e−2
26	.3..	7.91 e−2	3.90 e−2
13	.2..	7.37 e−2	3.46 e−2
10	..3.	6.67 e−2	3.03 e−2
23	..2.	6.50 e−2	3.03 e−2
42	.1..	6.25 e−2	3.03 e−2
8	.1..	6.25 e−2	3.03 e−2
15	..3.	5.69 e−2	2.60 e−2
16	..3.	5.69 e−2	2.60 e−2
33	..2.	5.54 e−2	2.60 e−2
29	..2.	5.54 e−2	2.60 e−2
1	.0..	5.40 e−2	2.60 e−2
18	..1.	5.20 e−2	2.60 e−2
43	.2..	4.53 e−2	2.16 e−2
55	..3.	3.75 e−2	1.73 e−2
38	..2.	3.66 e−2	1.73 e−2
49 50	..2.	1.81 e−2	8.66 e−3
41 41 17	.0..	8.78 e−3	4.33 e−3
49 31 49	.2..	3.93 e−3	8.66 e−3
41 41	.0..	3.63 e−3	4.33 e−3
49 49	..3.	2.19 e−3	4.33 e−3
17 4	..1.	1.68 e−3	8.66 e−3
17 4	..2.	1.09 e−3	1.73 e−2
60 60 17	..1.	1.06 e−3	1.30 e−2
4	..2.	8.24 e−4	3.46 e−2
17 4	.0..	8.20 e−4	8.66 e−3
17 17	..3.	7.76 e−4	2.16 e−2
41	..0.	7.37 e−4	6.49 e−2
49 49	..0.	7.04 e−4	4.33 e−3
49 41 49	.1..	7.03 e−4	8.66 e−3
28	.2..	6.67 e−4	2.16 e−2
49 60	.0..	6.15 e−4	4.33 e−3
11 40	..2.	5.59 e−4	1.73 e−2
41 40	.3..	5.14 e−4	4.33 e−3
41 49	..1.	4.97 e−4	1.30 e−2
17 4	.3..	4.97 e−4	8.66 e−3

Table 3: First 60 entries of inventory with a larger degree of fusionality (multiple atomic meanings per morpheme).

lence[6] of the morpheme (how often it occurred in the corpus during induction).

Inventory size Number of morphemes in the inventory.

Vocabulary size Number of distinct form tokens.

Inventory entropy Entropy of morphemes by normalized prevalence.

Form length Mean length of the morphemes' forms.

Meaning size Mean size of the morphemes' meanings.

Morpheme bijectivity Defined in Section 4.4.

Forms per meaning Mean number of distinct forms which map to a given meaning.

Meanings per form Mean number of distinct meanings which map to a given form.

Synonymy entropy Entropy of forms conditioned on a particular meaning (across all meanings).

Polysemy entropy Entropy of meanings conditioned on a particular form (across all forms).

F Environment Ablation Results

The full results for the ablation of the morpheme induction algorithm and phrasebook agents are shown in Figs. 8 and 9.

G Phrasebook Agent Ablation Details

Weight method The default method of weighting morphemes in CSAR is mutual information. Other methods include provided and tested include:

NPMI Normalized pointwise mutual information of the form and meaning.

Joint probability The probability of the form and meaning occurring together in the corpus.

PMIM Pointwise mutual information mass; the PMI of the form and meaning weighted by the their joint probability.

Applicability Defined as weighted probability of the meaning occurring given the form less the probability of the meaning not occurring given the form and the meaning occurring given the form is *not* present; i.e., $p(m,f)p(m|f) - p(\neg m,f)p(\neg m|f) - p(m,\neg f)p(m|\neg f)$.

Results The full results for the ablation of the morpheme induction algorithm and phrasebook agents are shown in Figs. 10 and 11.

[6] Raw prevalences do not add up to 1 since multiple morphemes can occur in one message, thus we normalize them to sum to 1 to treat them as a probability measure.

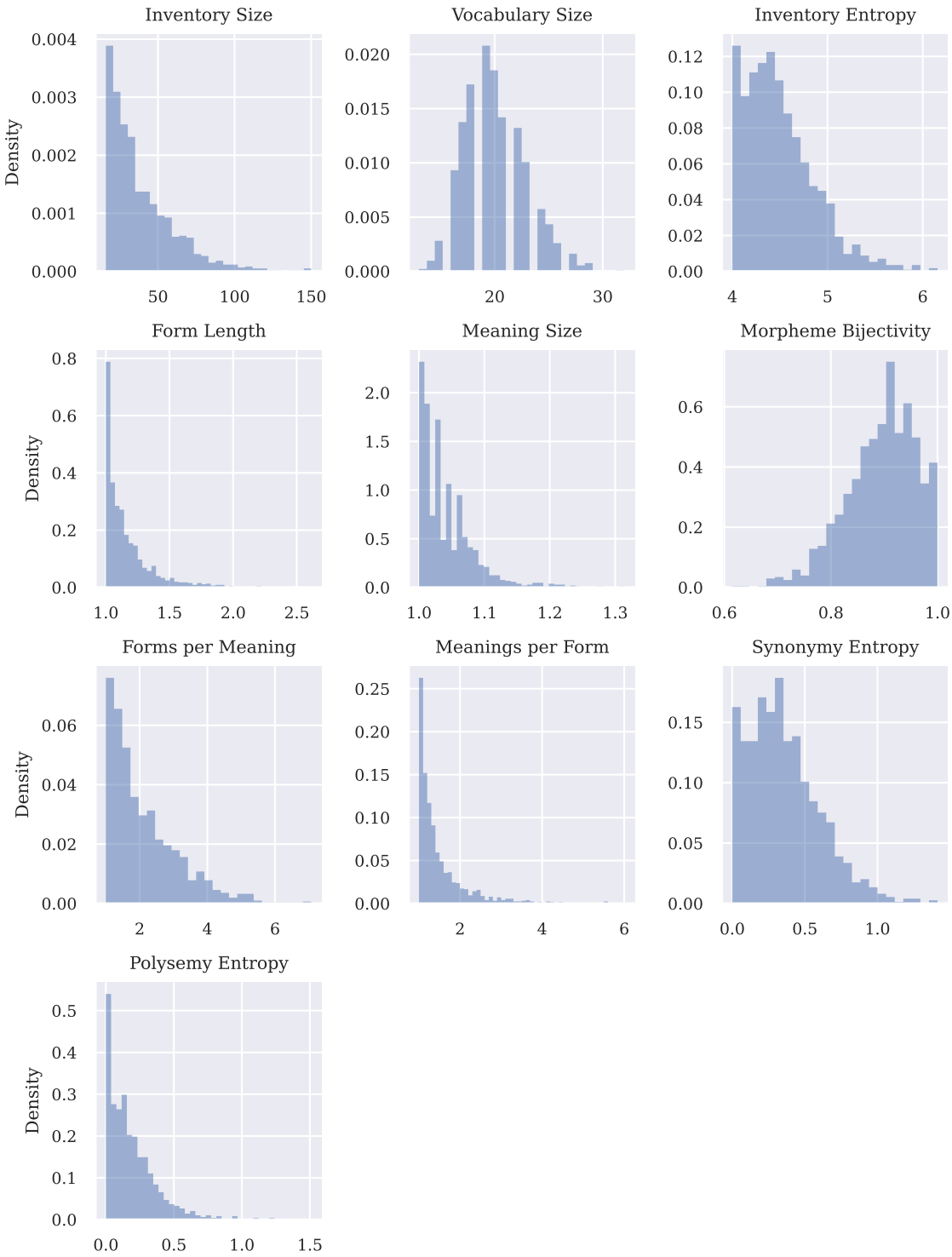


Figure 6: Histograms of metrics from main experiment.

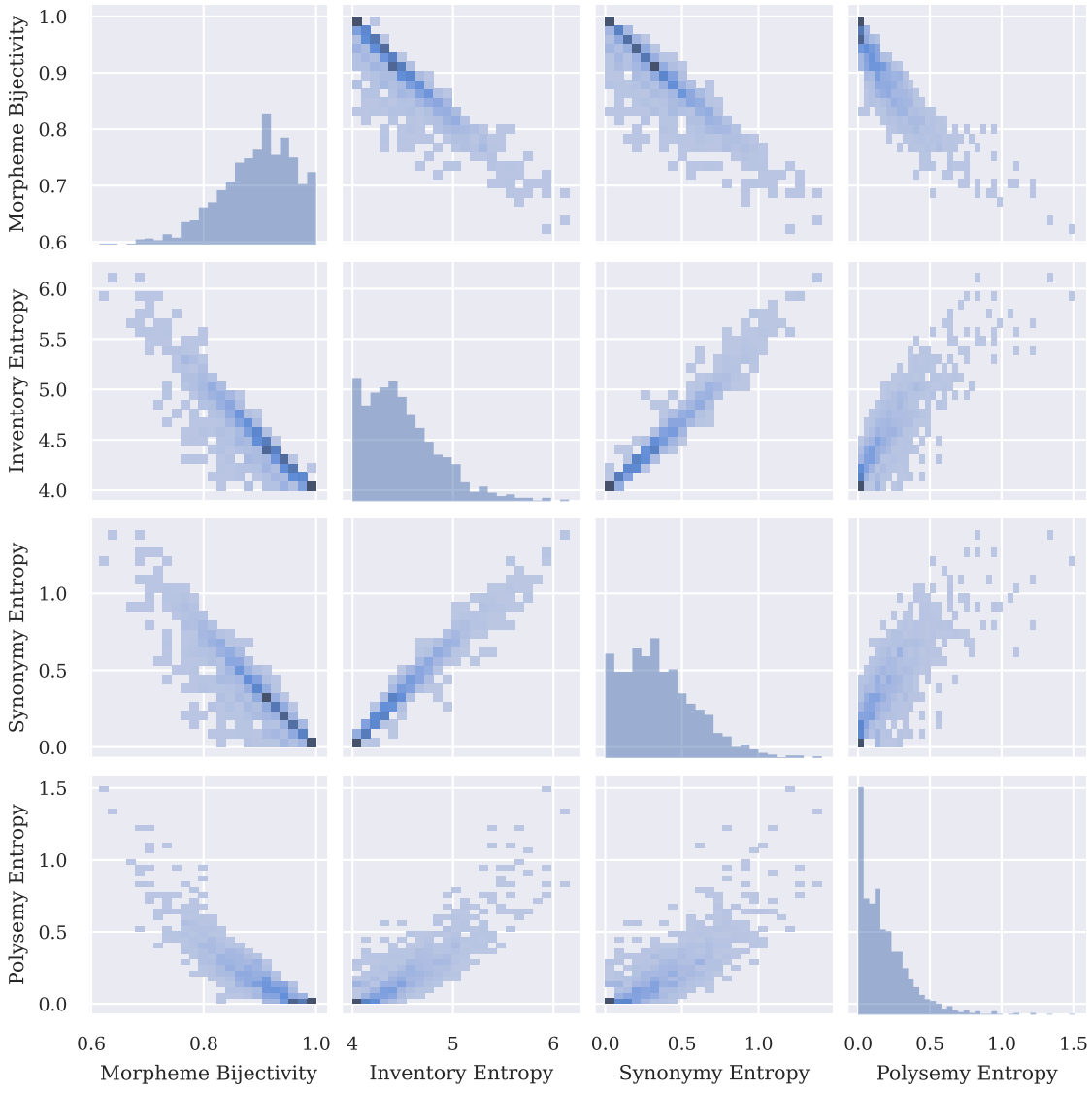


Figure 7: Bivariate histograms of selection of metrics from main experiment.

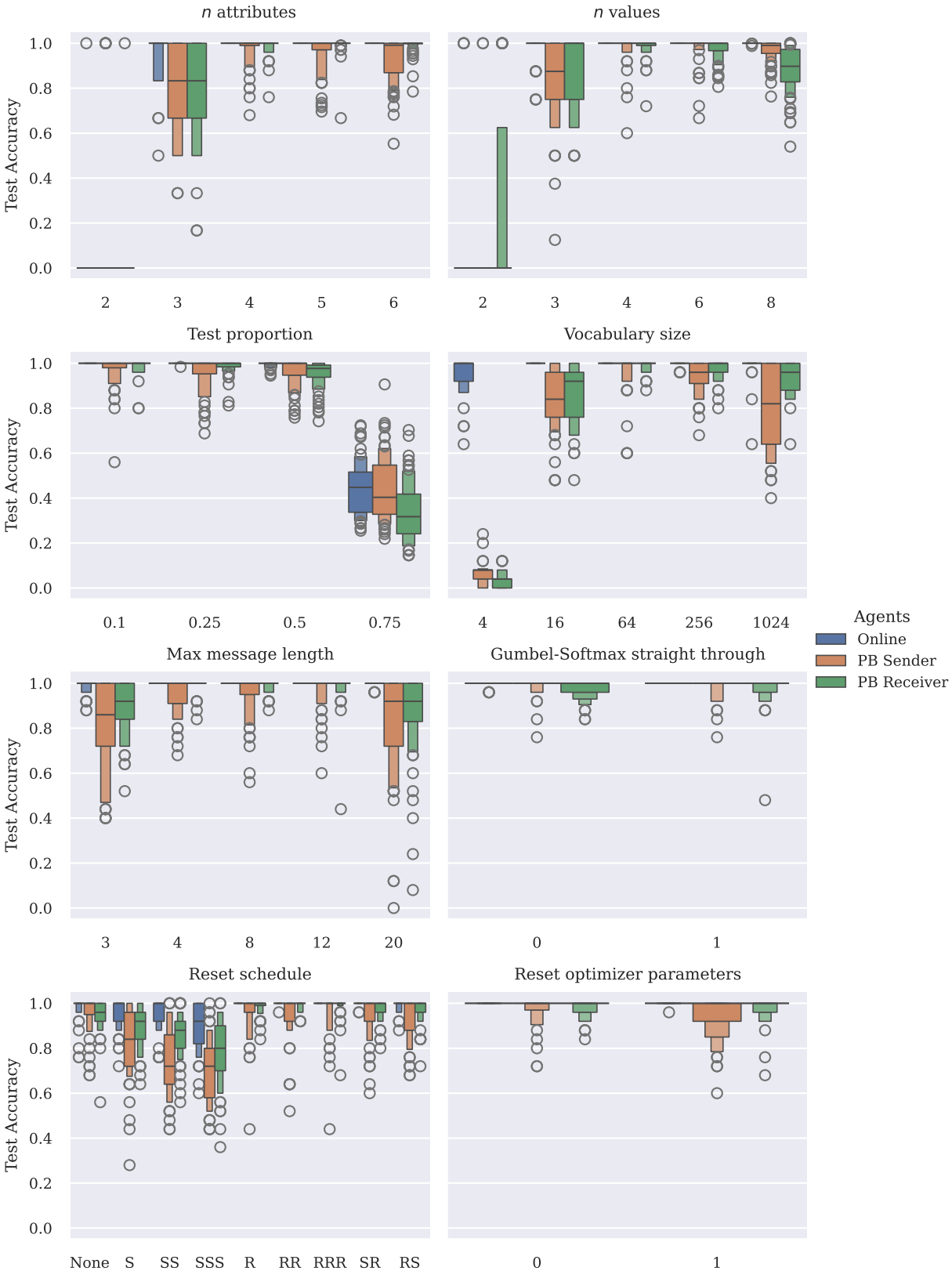


Figure 8: Phrasebook agents' test accuracy across morpheme induction and phrasebook agent ablations. *Default values.

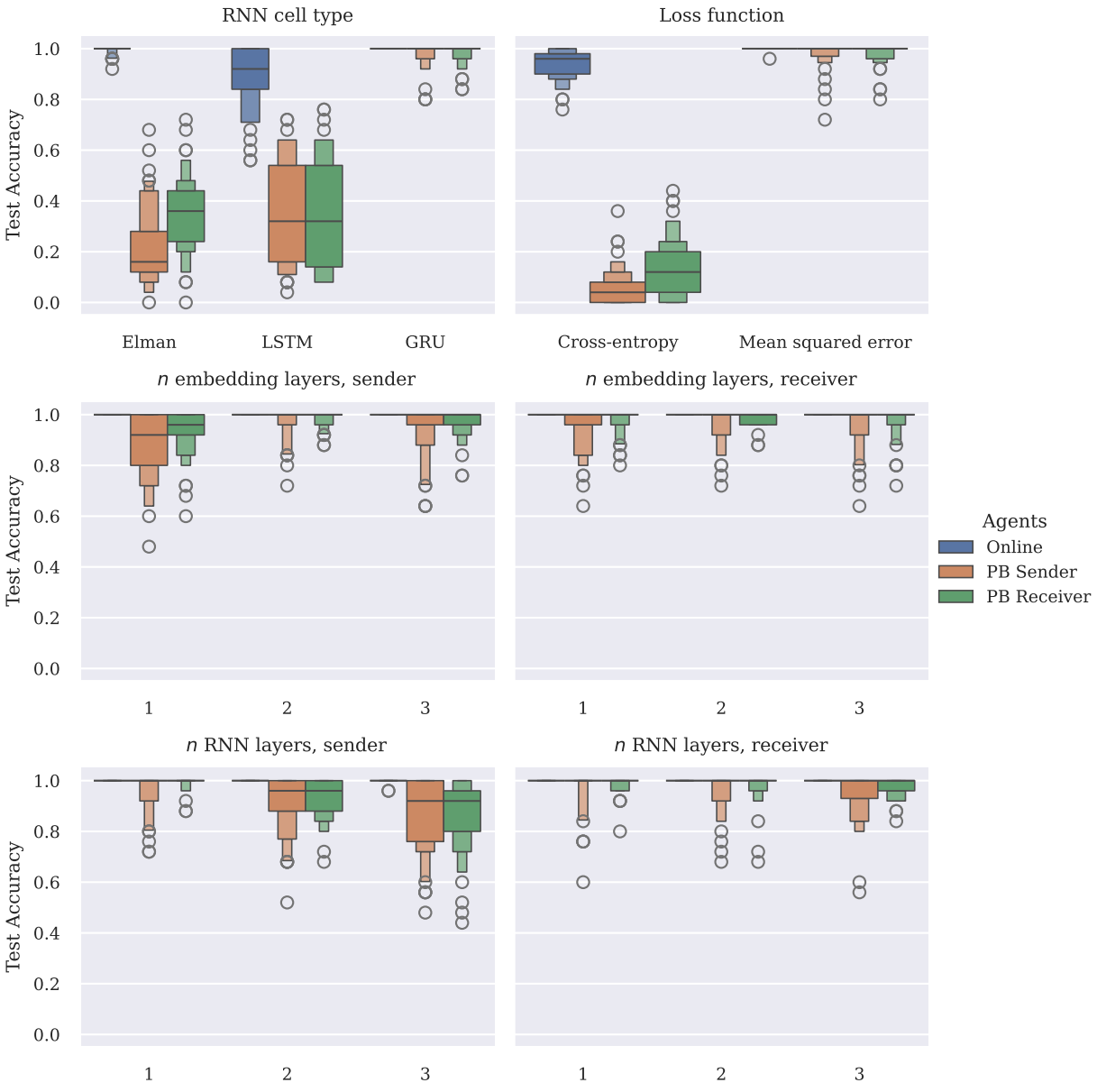


Figure 9: Phrasebook agents' test accuracy across morpheme induction and phrasebook agent ablations. *Default values.

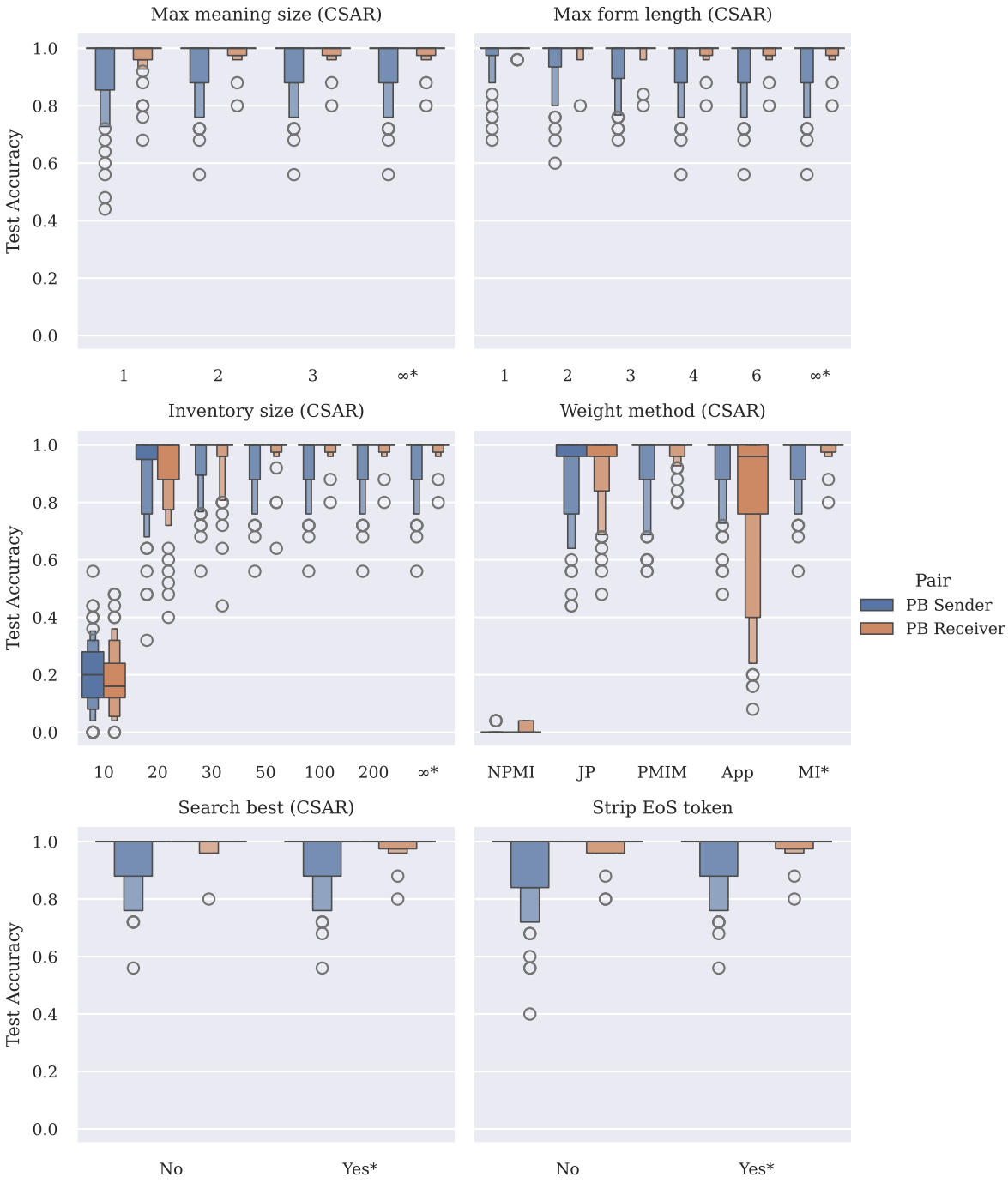


Figure 10: Phrasebook agents' test accuracy across morpheme induction and phrasebook agent ablations. *Default values.

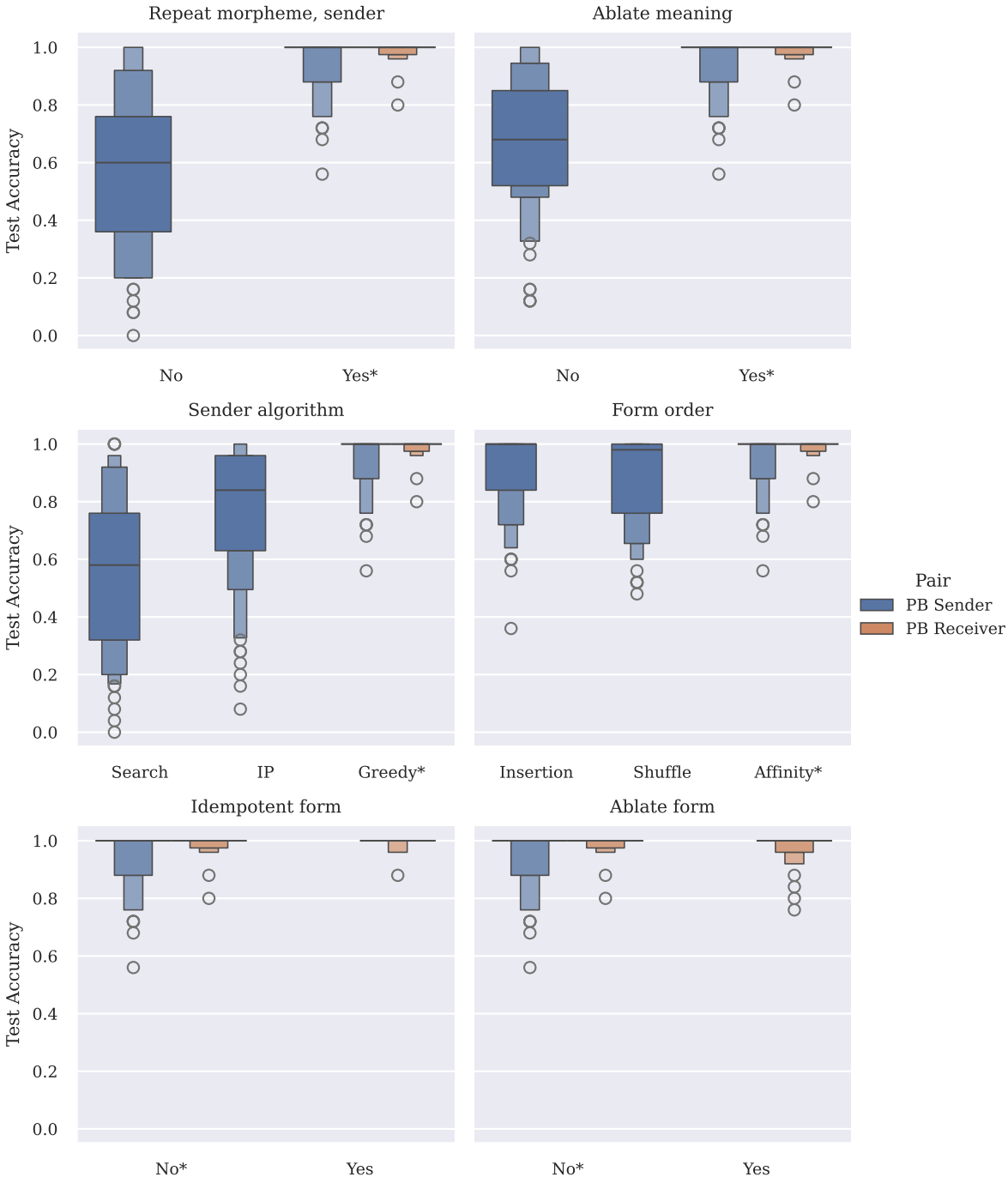
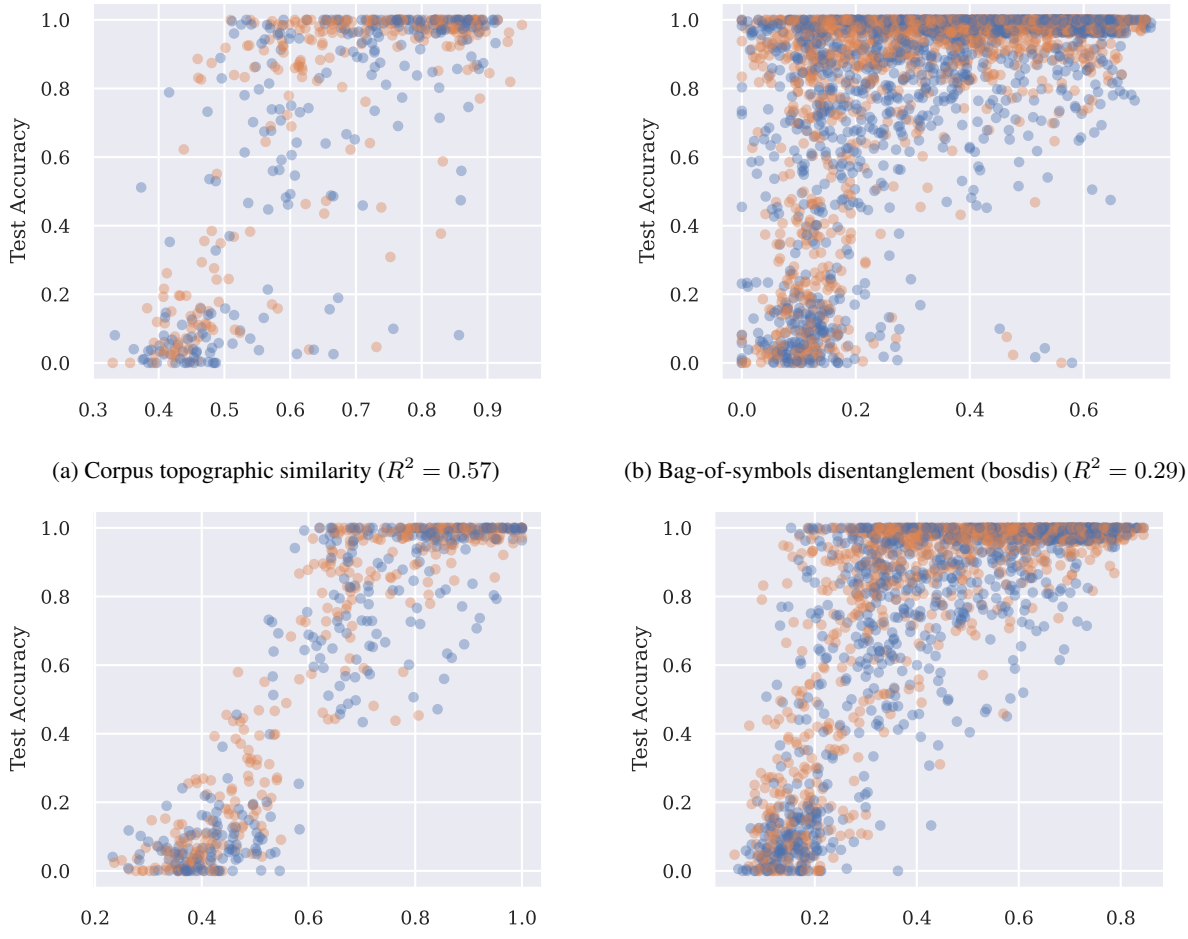


Figure 11: Phrasebook agents' test accuracy across morpheme induction and phrasebook agent ablations. *Default values.

H Compositionality Metrics

Scatter plots and R^2 values given in Fig. 12.



(a) Corpus topographic similarity ($R^2 = 0.57$)

(b) Bag-of-symbols disentanglement (bosdis) ($R^2 = 0.29$)

(c) Morpheme bijectivity ($R^2 = 0.85$)

(d) Morpheme mutual information ($R^2 = 0.54$)

Figure 12: Plot of different compositionality metrics vs. phrasebook sender (blue) and receiver (orange) accuracy. Note the difference in number of points is caused by stratified sampling which keeps a constant number of values per bucket on the x-axis.