n-dimensional system, and $\mathbf{L} = \mathbf{R}^{-1}$. Each eigenvalue λ_i has an associated eigenvector $\mathbf{r}_i$, which corresponds to a row of the matrix $\mathbf{R}$.

From this decomposition, the state of a n-dimensional linear dynamical system can be expressed as the linear composition of n independent one-dimensional exponential dynamics, often referred to as *modes* or patterns of activity. Each mode evolves along a direction in the state space defined by the eigenvector $\mathbf{v}_i$, which represents an invariant line in the phase space. Consequently, the motion near $\mathbf{h}^*$ can be understood as the linear combination of these n one-dimensional systems. The dynamic of a mode along the direction $\mathbf{v}_i$ is controlled by its associated eigenvalue λ_i, and the evolution of the system along that direction is given by $\lambda_i^t b_i$, where $\mathbf{b}_i$ is the initial amplitude of the mode. The magnitude of $|\lambda_i|$ determines the stability of the motion along $\mathbf{v}_i$. If $|\lambda_i| > 1$, the component of $\mathbf{h}_t$ along $\mathbf{v}_i$ grows exponentially, indicating an unstable direction. In contrast, if $|\lambda_i| < 1$, the component shrinks exponentially, which indicates a stable direction. The overall stability of the fixed point $\mathbf{h}^*$ is determined by the *spectral radius* of the Jacobian matrix, which is the largest magnitude among the eigenvalues of $\mathbf{JF}(\mathbf{h}^*)$. Specifically, if all the eigenvalues $|\lambda_i|$ are within the unit circle, the fixed point $\mathbf{h}^*$ is stable. If at least one eigenvalue satisfies $|\lambda_i| > 1$, the fixed point is unstable. For saddle points, some eigenvalues lie within the unit circle, while others lie outside, resulting in both stable and unstable directions. Finally, if all eigenvalues λ_i lie outside the unit circle, $\mathbf{h}^*$ is a totally unstable equilibrium point.

3.4. Basins of Attraction and Saddle Points

In many systems without external input, the dynamics naturally evolves toward specific regions of the state space. These converging points or regions are known as *attractors*. Among the attractors, the simplest type is the stable fixed point. The region of the phase space (i.e. the set of all initial states) from which the system evolves toward a particular attractor is called the *basin of attraction* of an attractor. Any initial condition within this region will eventually lead the system to the attractor through the iterative dynamics of the system. The state space is typically partitioned into basins of attraction, each associated with a specific attractor. Saddle points play a crucial role in governing the boundaries and interactions between these basins. Typically, a saddle point will have a dominant set of stable modes (or manifolds) with only a small number of unstable modes. This configuration makes saddle points critical for state space management, as they influence how trajectories transition between basins of attraction. For example, a region of state space may be funneled through the stable modes of a saddle point, only to be directed toward different attractors by its unstable modes. In this way, saddle points act as gateways or intermediaries that connect different basins of attraction. As a result of this interaction, the stable manifold of a saddle point often forms the boundary between the basins of attraction [8]. The complexity of a saddle point can be quantified by its *index*, defined as the number of unstable manifolds (or directions) associated with the fixed point. A saddle point with a higher index has a greater number of directions in which the trajectories diverge, which can lead to more intricate dynamics and transitions between basins.

3.5. Reverse Engineering RNNs of Classification Tasks

Recurrent Neural Networks, as nonlinear discrete-time dynamical systems, can be analyzed using tools from dynamical system theory. Modern RNN architectures are made up of hidden layers with hundreds of neurons, resulting in high-dimensional hidden states. Traditional dynamical system analysis often treats individual neurons as system parameters, but this high dimensionality poses significant challenges for standard state-space analysis. A recent line of research adopts a higher-level perspective to study the computational mechanisms learned by RNNs [55]. These reverse engineering techniques aim to uncover how trained RNNs implement specific tasks by analyzing the geometry and dynamics of their state space. This approach focuses on key dynamical features, such as fixed points, their linearized dynamics, and the interactions between equilibrium points, to infer the behavior of the network.

For example, tasks such as binary sentiment analysis and general text classification exhibit a common underlying dynamical mechanism [3]. In such tasks, the hidden state trajectories of RNNs largely lie in a low-dimensional subspace of the full state space, despite the high dimensionality of the hidden states. Within this subspace lies an attractor manifold, which serves as a repository for accumulating evidence for each class as the network processes tokens sequentially. The exact dimensionality and geometry of this attractor manifold depend on the structure of

the dataset and the complexity of the task. For binary sentiment classification, hidden states typically evolve along a line of stable fixed points, reflecting the network's progression as it processes input text [39]. More generally, for categorical classification tasks with N classes, the attractors form an $(N-1)$-dimensional simplex that captures the scalar quantities that the network must maintain to accurately classify the input [3].

4. Objectives

RNNs are nonlinear dynamical systems with high-dimensional state-space structures. These structures can be analyzed by examining fixed points, attractor manifolds, and trajectories within the state space. Recent advances in reverse engineering techniques have provided valuable insight into how RNNs implement task-specific computations. These studies suggest that RNNs encode evidence in low-dimensional manifolds, using an integrative mechanism to track and accumulate information over time to facilitate accurate classification. Although significant progress has been made in understanding RNN dynamics for tasks such as binary sentiment classification and generic text classification, these techniques have not yet been systematically applied to intent detection. Intent detection presents unique challenges due to its diverse and semantically rich set of intent classes. As a result, it remains unclear how state-space dynamics manifest in intent detection tasks and how architectural parameters affect the underlying trajectories and manifold geometry. The primary objective of this study is to investigate how RNN architectures encode intent detection tasks in their state space. Specifically, our goal is to analyze the structure of the state space, determining its intrinsic dimensionality, and comparing it with previous findings for generic categorical text classification tasks. We also examine the behavior of hidden state trajectories as the RNN processes input sequences, uncovering the internal mechanism that lead to accurate predictions. To address these objectives, we performed experiments using the SNIPS and ATIS datasets and analyzed the state-space dynamics of various RNN architectures.

5. Datasets

Several benchmark datasets have been widely used to evaluate the performance of intent detection models. Among them, three prominent datasets are SNIPS, ATIS, and MASSIVE, each with distinct characteristics and challenges. The SNIPS dataset [10] developed for English voice assistant systems consists of 7 balanced intents. This simplicity, combined with the absence of significant class imbalance, makes SNIPS an ideal choice for analyzing the state space dynamics of intent detection models without introducing side effects due to imbalance or an excessive number of classes. In contrast, the ATIS (Airline Travel Information System) dataset [21] contains real customer conversations about flight information. Although ATIS includes 26 intentions, almost 74% of its samples belong to a single intent, making it challenging to analyze model performance fairly across all classes. Furthermore, some utterances in ATIS are annotated with multiple intents. The recently introduced MASSIVE dataset [13] is a multilingual localization of the SLURP dataset [5]. It spans 51 languages and includes 60 intents in 18 domains. However, the MASSIVE dataset exhibits both a large number of intents and a degree of class imbalance, with some intents having very few samples. This makes MASSIVE more suitable for multilingual and large-scale evaluations.

Given our focus on understanding the state space dynamics of RNNs in intent detection tasks, we select two complementary datasets for our experiments: SNIPS and ATIS. SNIPS serves as our baseline. Its reduced number of intents, balanced class distribution, and manageable complexity allow for a controlled investigation of the underlying computational mechanisms. ATIS serves as our generalizability test. Its significant class imbalance and larger number of intents provides a challenging, real-world scenario to validate our framework. Table 1 compares the key characteristics of these datasets, highlighting differences in class distribution, imbalance, and scale.

Table 1
Comparison of intent detection datasets, summarizing key characteristics, including the number of languages, utterances per language, domains, intents, slots, and class imbalance measures (number and percentage of samples in the largest and smallest intent classes for each dataset).

Dataset Name	Languages #	Utterances per Language	Domains #	Intents #	Slots #	Samples in Largest Intent	Largest Intent (%)	Samples in Smallest Intent	Smallest Intent (%)
SNIPS	1	14484	-	7	53	2100	14.5	2042	14.1
ATIS	1	5871	1	26	129	4298	73.7	1	0.02
MASSIVE	51	19521	18	60	55	1190	6.9	6	0.04

6. Experiments Setup

Our analysis is structured into four steps: a) train different RNN architectures to solve the intent detection task on the SNIPS and ATIS datasets; b) extract and visualize the state space learned by the RNNs; c) analyze the manifold structure in which the state space is embedded; and d) investigate the fixed-point structure underlying the RNNs.

We implemented and trained the RNN models using TensorFlow 2 [1]. Each RNN consisted of a trainable embedding layer with *embed_dim* neurons (without pre-trained embeddings), a unidirectional recurrent layer with *hidden_dim* neurons, and a final dense layer for output. Tokenization was performed using Tensorflow's *TextVectorization* layer. Three types of recurrent cells were evaluated: standard (vanilla) RNN, LSTM [22], and GRU [9]. The models were trained on the SNIPS and ATIS intent detection dataset. The training process optimized the multiclass cross-entropy loss function, used for classification tasks. We used Adam optimizer [31] for all experiments. Training was performed with early stopping based on validation accuracy [42] (patience=2 epochs) to prevent overfitting.

It is important to note that our goal was not to achieve state-of-art optimized performance, but rather to obtain a reasonably high-performance (e.g., > 93% accuracy). This ensures the models are competent and their learned dynamics, which are the focus of our study, are meaningful. To this end, hyperparameters were tuned independently for each dataset. For the SNIPS dataset we used a learning rate of $\eta = 5 \times 10^{-4}$ and a batch size of 32. No additional regularization was applied. For the ATIS dataset, we used a learning rate of $\eta = 2.5 \times 10^{-3}$ which was halved after the second epoch. and a batch size of 16. To manage this dataset's complexity and achieve our target performance, we also applied dropout (rate=0.2) [52], to the recurrent layer. The best performing model for each architecture was selected based on its performance in a validation dataset. The validation subsets were created by randomly sampling 20% of the training data, to ensure a balanced representation of the target classes. The balanced class distribution of the SNIPS dataset made accuracy an appropriate evaluation metric. For the imbalanced ATIS dataset, we report disaggregated F1-scores to provide a more nuanced analysis. All metrics were calculated on a separate test dataset not used during training or validation. Each training procedure was repeated using 10 different seeds.

7. Results

7.1. Intent Detection Low-dimensional Dynamics

In this section, we show that the state space learned by an RNN during the intent detection task is constrained to a low-dimensional hypersurface, or manifold, embedded within the high-dimensional space of the hidden layer. Before sentences are processed by RNNs, a tokenization mechanism transforms each natural language phrase into a sequence $\mathbf{x}_1, \ldots, \mathbf{x}_T$, where each token $\mathbf{x}_i \in \mathbb{R}^m$ is an m-dimensional vector and T is the number of tokens in sentence [26]. As shown in Figure 2, injecting this sequence into an RNN generates a corresponding sequence of activations $\mathbf{h}_1, \ldots, \mathbf{h}_T$ in the hidden layer. Each hidden state $\mathbf{h}_i \in \mathbb{R}^n$ is an n-dimensional vector (with $n = \textit{hidden\_dim}$) computed using Equation 1. The collection of all hidden states visited by the input sentences constitutes the state space learned by the trained RNN.

Given that the state space points may lie on a manifold embedded within a higher-dimensional space, the natural question is to determine the intrinsic dimensionality of this manifold. Intrinsic dimensionality refers to the minimum number of dimensions required to accurately represent the variability of the data. Several methods exist to estimate