# Semantic Approximation Theory via Transformers: A Partial Order-Based Framework

Author

December 27, 2026

**Abstract**

This paper presents a rigorous theoretical framework for modeling the relationship between texts and concepts using partial order structures. We argue that traditional euclidean representations in Natural Language Processing (NLP) have fundamental limitations in capturing the hierarchical and deductive relationships inherent in human language. We propose structured spaces of texts ($\mathcal{T}$) and concepts ($\mathcal{C}$) endowed with partial orders that reflect semantic specificity. We demonstrate that both spaces can be completed to form algebraic domains, and that the Lawson topology provides a suitable notion of convergence for semantic approximations. Our main result establishes that any Scott-continuous semantic function can be approximated arbitrarily closely by Transformer architectures. This work lays the mathematical foundations for developing language models with structured reasoning capabilities.

**Keywords:** Formal semantics, partial orders, algebraic domains, Lawson topology, Transformers, universal approximation, natural language processing.

## 1 Introduction

Artificial intelligence has achieved extraordinary milestones in natural language processing, but a fundamental question persists: are we capturing the true essence of meaning or merely learning superficial patterns? In recent years, the community has recognized that progress is no longer measured primarily by the scale of models or their predictive capacity, but by their ability to reason and genuinely understand. Modern transformers generate fluent and coherent text, but their understanding of meaning remains an enigma. The problem lies in the fact that our current representations do not align with the inherent structure of human language.

The era of “bigger is better” is giving way to a new phase where the quality of reasoning matters more than the number of parameters. While massive models can memorize complex patterns, their capacity for structured and comprehensive reasoning remains limited. This transition from purely predictive models to systems with genuine reasoning capabilities demands representations of meaning that capture the logical and conceptual relationships underlying language.

Traditional approaches based on vector embeddings, though powerful for predictive tasks, prove insufficient for modeling structured reasoning. By mapping texts to points in a metric space, we lose the order relationships that are intrinsic to meaning and essential for logical reasoning.

Human language is not a flat collection of symbols, but a deeply structured system where words and phrases are related through natural conceptual hierarchies. When we say "domestic cat," we implicitly understand that it is a specific type of "cat," which in turn is a "feline," which is a "mammal," and so on. This hierarchical structure is not accidental—it is fundamental to how humans organize and communicate knowledge.

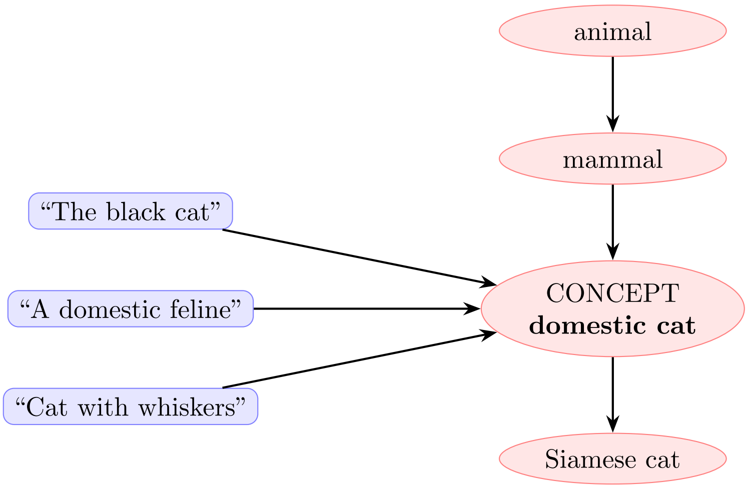


Figure 1: Unified conceptual structure unifying diverse linguistic expressions

## 1.1 Limitations of Metric Approaches

Traditional embeddings represent texts as vectors in high-dimensional spaces, where semantic similarity is measured via distances. While this approach has enabled significant advances, it presents fundamental limitations when attempting to capture the true structure of meaning:

- **Loss of hierarchical relationships**: Distance metrics cannot naturally express that "cat" is a specific type of "animal."
- **Artificial separation of equivalents**: "gato" (Spanish) and "cat" (English) may end up in very different regions of the vector space.
- **Lack of structural compositionality**: There are no guarantees that combining embeddings preserves semantic relationships.
- **Difficulty in modeling reasoning**: Trajectories in vector space do not correspond to natural deductive chains.

These limitations are not merely technical—they reflect a fundamental mismatch between computational representation and the real nature of language.

## 1.2 Partial Order as a Natural Linguistic Structure

We propose that language inherently possesses a partial order structure, where generalization and specification relationships capture essential semantic connections. This perspective allows us to model meaning in a way that naturally aligns with how humans understand and use language.

We define orders between texts and concepts as:

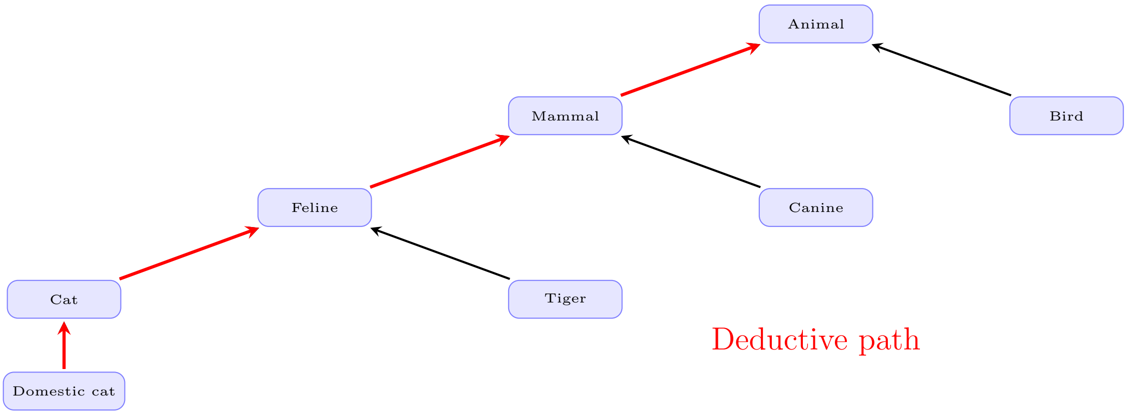
$$t_1 \leq_T t_2 \iff t_1 \text{ is a prefix of } t_2.$$

For example: "The cat" $\leq_T$ "The cat and the dog are friends."

$$c_1 \leq_C c_2 \iff c_1 \text{ encompasses } c_2$$

For example: "animal" $\leq_C$ "mammal" $\leq_C$ "feline" $\leq_C$ "cat."

One of the most powerful advantages of the partial order approach is how it naturally models reasoning chains. Instead of being arbitrary sequences of texts, deductions appear as structured paths through the conceptual space:



Each step in reasoning corresponds to moving in the ordered structure, either toward greater generalization or greater specification. This natural correspondence between linguistic structure and deductive process is a fundamental advantage over metric approaches.

The conceptual partial order provides a unified framework that transcends linguistic particularities. While traditional embeddings are strongly coupled to the training language, the ordered structure is universal:

- **Cross-linguistic equivalences**: "gato," "cat," "chat" map to the same ordered concept.
- **Formulation independence**: Different ways of expressing the same concept converge.
- **Preservation of relationships**: Conceptual hierarchies are maintained.

This adaptability makes the approach particularly suitable for multilingual applications and for modeling deep semantic understanding.

### 1.3 The Ideal Meaning Function

The central objective of our work is to learn the meaning function:

$$F : \mathcal{T} \to \mathcal{C}$$

that assigns to each text $t \in \mathcal{T}$ its corresponding conceptual meaning $c \in \mathcal{C}$. This function must satisfy fundamental properties that reflect how we understand meaning:

1. **Monotonicity**: If $t_1 \leq_T t_2$ then $F(t_1) \leq_C F(t_2)$.

   *Interpretation*: Adding information to a text produces more specific concepts, never more general ones. For example, if we have "The cat" $\leq_T$ "The old black cat chases a bird," then this adds information to the meaning, $F(\text{"cat"}) \leq_C F(\text{"black cat"})$.

2. **Preservation of equivalences**: Texts that express the same concept must map to the same conceptual element.

3. **Continuity**: Small changes in textual formulation produce small changes in the resulting concept.

4. **Multilingualism**: The conceptual structure is independent of the language of expression.

These properties capture the essence of how meaning is structured in human language: in an ordered, compositional, and formulation-independent manner.

The theoretical core of our work demonstrates that it is possible to learn the meaning function $F$ with the desired properties. Specifically, we prove that:

**Theorem 1** (Universal Approximation of the Meaning Function)**.** *Let $\mathcal{T}$ be the space of texts with the order $\leq_T$ and $\mathcal{C}$ the space of concepts with the order $\leq_C$. For every meaning function $F : \mathcal{T} \to \mathcal{C}$, there exists a transformer network that approximates $F$.*

This result establishes that we can not only learn to map texts to concepts, but we can do so while preserving the essential structure of meaning—the monotonicity that guarantees that adding information leads to greater specification, the preservation of equivalences that captures the notion of synonymy, and the continuity that ensures robustness against formulation variations.

## 2 Related Work

**Semantic representations in NLP.** Most contemporary approaches to natural language processing rely on vector representations of text, where semantic similarity is modeled through distances in high-dimensional metric spaces. Early distributed representations such as Word2Vec [6] and GloVe [8], as well as contextual embeddings produced by Transformer-based models such as BERT [3], have demonstrated remarkable empirical success. However, these approaches encode semantics within a metric framework that does not naturally capture hierarchical relationships, semantic specificity, or deductive structure. Notions such as entailment or refinement are only indirectly approximated and lack formal guarantees. Our work departs from this paradigm by modeling meaning through partial orders, where semantic structure is explicit rather than emergent.

**Formal semantics and structured meaning.** The view that meaning possesses an intrinsic logical and compositional structure has a long tradition in formal semantics. Montague semantics [7] provided a rigorous foundation for compositional meaning, while later developments such as situation semantics [2] and conceptual spaces [4] emphasized relational and geometric organization of concepts. While these frameworks successfully formalize semantic relationships, they are not designed to address approximation, learnability, or implementation by modern neural architectures. In contrast, our approach connects formal semantic structure with approximation theory and neural realizability.

**Order-theoretic and domain-theoretic approaches.** Partial orders and domain theory have played a central role in the semantics of computation, particularly in denotational semantics of programming languages. Complete partial orders, Scott-continuous functions, and algebraic domains provide a principled framework for modeling computation via finite information [9, 1, 5]. Although these tools are well established in theoretical computer science, their application to natural language semantics and representation learning has received limited attention. Our work adapts domain-theoretic ideas to linguistic objects, interpreting texts and concepts as elements of algebraic domains and semantic interpretation as a Scott-continuous map between them.

**Approximation theory and Transformers.** Recent work has established that Transformer architectures possess strong expressive power, including universal approximation properties for classes of functions on sequences [10]. These results are typically formulated in metric or measure-theoretic settings and do not account for semantic structure. Our universal approximation theorem differs in nature: it characterizes Transformers as dense in a space of Scott-continuous semantic functions endowed with the Lawson topology. This positions Transformers not merely as powerful sequence models, but as architectures capable of approximating structured semantic interpretations that respect order, hierarchy, and deduction.

## 2.1 Contributions

This work makes the following contributions:

- **Order-theoretic model of meaning.** We propose a semantic framework in which texts and concepts are modeled as partially ordered sets, where the order encodes semantic specificity and deductive refinement, providing a structural alternative to metric embedding spaces.
- **Algebraic domain structure for texts and concepts.** We show that both the space of texts and the space of concepts can be completed into algebraic domains, with finite texts and finite concepts corresponding to

compact elements. This enables principled finite approximations of meaning.

- **Universal approximation theorem via Transformers.** We prove that Transformer architectures can approximate any Scott-continuous semantic function arbitrarily well in the Lawson topology, providing a structural universal approximation theorem for language models.

# 3 Axiomatic costruction of the space of texts and concepts

## 3.1 Partial Orders and Specificity Structures

**Definition 2** (Partial Order). *A* partial order *is a pair $(P, \leq)$ where $P$ is a set and $\leq$ is a binary relation that satisfies:*

1. ***Reflexivity:*** $\forall x \in P,\ x \leq x$
2. ***Antisymmetry:*** $\forall x, y \in P,\ (x \leq y \wedge y \leq x) \Rightarrow x = y$
3. ***Transitivity:*** $\forall x, y, z \in P,\ (x \leq y \wedge y \leq z) \Rightarrow x \leq z$

### 3.1.1 Text Space $\mathcal{T}$

**Definition 3** (Text Space $\mathcal{T}$). *Let $V$ be a finite vocabulary. We define the text space as:*

$$\mathcal{T} = \{t : t \text{ is a finite sequence of elements of } V\}$$

*with the order $t_1 \leq t_2$ if $t_1$ is a prefix of $t_2$.*

**Example 4.**
- *"The cat" $\leq$ "the black cat eats tuna" (prefix)*
- *"the cat eats tuna" $\nleq$ "tuna eats the cat" (not a prefix)*

**Remark 5.** *One could consider other finer orders within texts, such as subtext or subsequence orders.*

**Remark 6** (Countability of $\mathcal{T}$). *$\mathcal{T}$ is countable, since it is the countable union $\bigcup_{n\in\mathbb{N}} V^n$, where each $V^n$ is the set of sequences of length $n$, which are finite sets.*

**Remark 7** (Finite below). *For every $t \in \mathcal{T}$, the set $\{s \in \mathcal{T} : s \leq t\}$ is finite.*

### 3.1.2 Concept Space $\mathcal{C}$

**Definition 8** (Concept Space $\mathcal{C}$). *Let $\mathcal{C}$ be the abstract set of semantic concepts. We define the order as:*

$$c_1 \leq c_2 \iff c_2 \text{ is more specific than } c_1$$

**Example 9.**
- *animal $\leq$ mammal $\leq$ cat (more specific)*
- *car $\nleq$ animal (not comparable)*

**Remark 10.** *Since each concept can be represented by at least one text, it follows that $\mathcal{C}$ is countable.*

**Remark 11.** *Note that for all $c_1, c_2 \in \mathcal{C}$, there exists $c_3 \geq c_1, c_3 \geq c_2$ (a concept more specific than both), for example $c_1 =$ "Car", $c_2 =$ "Medicine", $c_3 =$ "Ambulance".*

## 3.2 Directed Sets and Completeness

A notion of utmost importance in mathematics is that of **convergence**. The following definition seeks to generalize this concept.

**Definition 12** (Directed Set)**.** *A subset $D \subseteq P$ of a partial order is* directed *if:*

1. $D \neq \emptyset$
2. $\forall x, y \in D,\ \exists z \in D$ *such that* $x \leq z$ *and* $y \leq z$

**Proposition 13.** *Let $D$ be finite and directed. Then it has a maximum.*

*Proof.* By induction on the number of elements of $D$:

1. If $D$ has one or two elements, it is trivial.
2. Let $D$ have $n+1$ elements. Take $x, y \in D$. Since $D$ is directed, there exists $z \in D$ greater than both. Then, without loss of generality, $D \setminus \{x\}$ is directed because, by transitivity, all elements smaller than $x$ are smaller than $z$. Therefore, by the inductive hypothesis, there exists $\max(D \setminus \{x\}) \geq z$, which is also greater than $x$. Hence, it is the maximum of $D$.

□

**Proposition 14** (Characterization of Directed Sets in $\mathcal{T}$)**.** *A set $D \subseteq \mathcal{T}$ is directed if and only if there exists a sequence $t_1 \leq t_2 \leq t_3 \leq \cdots$ in $D$ such that every element of $D$ is less than or equal to some $t_i$.*

*Proof.* ($\Rightarrow$) Let $D$ be directed. We show that $D$ is totally ordered. Given $s, t \in D$, there exists $u$ such that $s$ and $t$ are prefixes of $u$. Then the one with shorter length will be a prefix of the other. Moreover, by antisymmetry, this implies that there can be at most one text per length. Therefore, since $D \subset \mathcal{T}$ is countable and bounded below, we can represent $D$ as the sequence of texts enumerated by their length.

($\Leftarrow$) Every non-empty ascending chain is directed. Given $t_1, t_2$, it suffices to take their maximum. □

**Example 15** (Directed Sets in $\mathcal{T}$)**.**

- *$D_1 = \{$ "The", "The cat", "The cat eats"$\}$ (directed - ascending chain)*
- *$D_2 = \{$ "The", "cat"$\}$ (not directed - no text contains both as prefixes)*

We will define the limit as the supremum of a directed set in some form.

**Definition 16** (Supremum and Infimum)**.** *Let $(P, \leq)$ be a partial order and $S \subseteq P$:*

- *The* supremum *of $S$ (if it exists) is the least upper bound of $S$, denoted $\bigsqcup S$.*
- *The* infimum *of $S$ (if it exists) is the greatest lower bound of $S$, denoted $S$.*

**Remark 17.** *If $S$ has a maximum, then it is the supremum.*

**Remark 18** (Existence of suprema for finite sets in $\mathcal{C}$)**.** *Since for any two concepts $c_1, c_2$, one can find the concept $c_3 = c_1 \sqcup c_2$ that is the most specific concept that is more specific than both, i.e., the supremum. For example, for $c_1 =$ "Dogs", $c_2 =$ "Cats", and $c_3 =$ "Dogs and Cats". By induction on the number of elements in the set, it can be verified that for any finite subset $S \subset \mathcal{C}$, there exists $\bigsqcup S$, the supremum of $S$.*

**Definition 19** (CPO - Complete Partial Order)**.** *A partial order $(P, \leq)$ is a* CPO *if:*

1. *It has a minimum element $\perp$.*
2. *Every directed set $D \subseteq P$ has a supremum $\bigsqcup D$.*

**Remark 20.** *$\mathcal{T}$ is not a* CPO *because if we consider a set $D$ that is an infinite ascending chain, there is no supremum within finite texts.*

**Definition 21** (Completion of $\mathcal{T}$)**.** *We define the completion of $\mathcal{T}$ as $\overline{\mathcal{T}} = \mathcal{T} \cup \{\text{infinite sequences}\} \cup \{\varepsilon\}$, where $\varepsilon$ is the empty sequence and we consider the naturally extended order.*

**Theorem 22.** *$\overline{\mathcal{T}}$ is a CPO.*

*Proof.* Let $D \subseteq \overline{\mathcal{T}}$ be directed. First, if $D$ is finite, it has a maximum and the result is immediate. Otherwise, we claim that there can be at most one infinite sequence in $D$. Indeed, if there exist infinite sequences $s, t \in D$, then since $D$ is directed, there exists $u$ such that $s \leq u$ and $t \leq u$. But infinite sequences have no upper bounds other than themselves, so $s = u = t$. Therefore, if $D$ contains an infinite sequence, it is the maximum. If not, using the previous characterization, we have $t_1 \leq t_2 \leq \cdots$. We take $t_\infty$ as the sequence that has as its $i$-th word the common word for all $t_j$ with $j \geq i$. Clearly, this is an upper bound of the set and it is unique, hence it is the supremum. □

In fact, $\overline{\mathcal{T}}$ satisfies the universal property of being the unique CPO that contains $\mathcal{T}$ as a dense subset, up to isomorphism.

# 4 Algebraic Domains and Approximation Properties

In this section, we introduce a concept of fundamental importance for computational applications: compact elements. Spaces with a good relationship with their compact elements will allow us to work finitely.

**Definition 23** (Compact Element)**.** *An element $k \in P$ of a CPO is* compact *if for every directed set $D$ with $k \leq \bigsqcup D$, there exists $d \in D$ such that $k \leq d$.*

**Proposition 24** (Compact Elements in $\overline{\mathcal{T}}$)**.** *The compact elements of $\overline{\mathcal{T}}$ are exactly the finite texts.*

*Proof.* We separate cases according to whether $t$ is finite or infinite:

- If $t$ is finite and $t \leq \bigsqcup D$, then either $D$ has a maximum and therefore $t \leq \bigsqcup D \in D$, or $\bigsqcup D = t_\infty$ (an infinite sequence). In the latter case, $t$ is a prefix of $t_\infty$ of length $n$. Then we take any text $t_i$ from $D$ with $i \geq n$ and we have $t \leq t_i$.
- If $t$ is infinite, consider $D = \{\text{finite prefixes of } t\}$. Then $\bigsqcup D = t$ but no element of $D$ is greater than $t$.

□

**Definition 25** (Algebraic Domain)**.** *A CPO $(D, \leq)$ is an* algebraic domain *if:*

- *The set of compact elements $K(D)$ is countable.*
- *For every $x \in D$: $x = \bigsqcup\{k \in K(D) : k \leq x\}$.*

**Theorem 26** ($\overline{\mathcal{T}}$ is an Algebraic Domain)**.** *The completion $\overline{\mathcal{T}}$ is an algebraic domain.*

*Proof.* In a straightforward manner:

- $K(\overline{\mathcal{T}}) = \mathcal{T}$, which is countable.
- For every $t \in \overline{\mathcal{T}}$: $t = \bigsqcup\{\text{finite prefixes of } t\}$.

□

Thus, all information is contained in the compact elements, which are more manageable computationally.

# 5 Topologies on Partial Orders

To define approximations on partially ordered sets, we need to endow them with an appropriate topological structure. A first natural idea is to generalize closed intervals of $\mathbb{R}$, considering sets of the form $[a, b] = \{x \in D : a \leq x \leq b\}$. We could then say that $x_n \to x$ if for every interval containing $x$, the sequence eventually stays within it.

However, there are alternatives that better exploit the structure of algebraic domains, using instead of intervals the sets $\uparrow k = \{x \in D : k \leq x\}$ where $k$ is compact.

## 5.1 Scott Topology

**Definition 27** (Scott Convergence)**.** *A sequence $(x_n)$* converges to $x$ in the Scott topology *if for every compact element $k \leq x$, there exists $n_0$ such that for all $n \geq n_0$, we have $k \leq x_n$. Formally:*

$$\forall k \in K(D) \textit{ with } k \leq x,\ \exists n_0 \textit{ such that } \forall n \geq n_0,\ x_n \in \uparrow k$$

Functions that preserve this notion of convergence are particularly important:

**Definition 28** (Scott-Continuous Functions)**.** *A function $f : D \to E$ between CPOs is* Scott-continuous *if it preserves:*

1. ***Orders:*** *$x \leq y \Rightarrow f(x) \leq f(y)$.*
2. ***Directed suprema:*** *$f(\bigsqcup D) = \bigsqcup f(D)$ for $D$ directed.*

Unfortunately, the Scott topology presents serious limitations for modeling convergence in semantic contexts:

**Example 29** (Uniqueness Problem)**.** *The constant sequence "The", "The", . . . converges not only to "The", but also to "The cat", "The dog" and all its possible extensions. Limits are not unique.*

**Example 30** (Specificity Problem)**.** *The constant sequence "The black cat with white whiskers", . . . converges to "The cat", which does not adequately capture the intuitive notion of convergence.*

These examples show that the Scott topology is too coarse for our purposes, as it does not adequately distinguish between different levels of specificity.

For those familiar with general topology, Scott convergence corresponds to the topology generated by the base $\{\uparrow k : k \in K(D)\}$. In an algebraic domain, this topology is $T_0$ but not necessarily $T_1$.

## 5.2 Lawson Topology

To resolve these limitations, we introduce a finer topology:

**Definition 31** (Lawson Basic Neighborhood)**.** *In an algebraic domain $D$, a* Lawson basic neighborhood *of $x \in D$ is a set of the form:*

$$B(x) = (\uparrow k) \setminus (\uparrow k_1 \cup \cdots \cup \uparrow k_n)$$

*where:*

- *$k \in K(D)$ with $k \leq x$ (inclusion condition).*
- *$k_1, \ldots, k_n \in K(D)$ with $k_i \not\leq x$ for each $i$ (exclusion conditions).*

**Example 32** (Semantic Neighborhood)**.** *For $x$ = "domestic cat", a typical Lawson neighborhood would be:*

$$B(x) = \{t : \text{“cat”} \leq t\} \setminus \{t : \text{“wild”} \leq t \text{ or } \text{“sick”} \leq t\}$$

*This neighborhood includes texts that mention cats but excludes those that qualify them as wild or sick.*

**Definition 33** (Lawson Convergence)**.** *A sequence $(x_n)$* converges to $x$ in the Lawson topology *if for every Lawson basic neighborhood $B(x)$, there exists $n_0$ such that for all $n \geq n_0$, we have $x_n \in B(x)$.*

The Lawson topology overcomes the limitations of the Scott topology:

- **Unique limits**: Lawson convergence guarantees uniqueness of limits.
- **Sensitivity to specificity**: Adequately distinguishes between different levels of detail.
- **Separation properties**: It is Hausdorff, unlike the Scott topology.
- **Adequate balance**: Captures both positive information (what it must contain) and negative information (what it must exclude).

The Lawson topology is strictly finer than the Scott topology and is particularly suitable for modeling approximation processes in formal semantics, where we need to control both inclusion and exclusion of information.

# 6 Semantic Function and Structure Inheritance

We have already shown that the space of texts $\overline{\mathcal{T}}$ is an algebraic domain. Given the strong relationship between language and concepts, we would like to see that $\mathcal{C}$ also is. To this end, we assume something quite natural: there exists a function $F : \overline{\mathcal{T}} \to \mathcal{C}$ that interprets the meaning of each text and satisfies the following properties:

1. $F$ is surjective, since every concept can be represented by some text.
2. $t_1 \leq t_2 \Rightarrow F(t_1) \leq F(t_2)$, i.e., adding text makes the meaning more specific.
3. There exists $G : \mathcal{C} \to \overline{\mathcal{T}}$ Scott-continuous such that $F \circ G = \mathrm{id}_{\mathcal{C}}$, which means that each concept has a canonical textual representation.

**Theorem 34** (Inheritance of Algebraic Domain Structure)**.** *Let $\overline{\mathcal{T}}$ be an algebraic domain and let $F : \overline{\mathcal{T}} \to \mathcal{C}$ be a Scott-continuous surjective function, and let $G : \mathcal{C} \to \overline{\mathcal{T}}$ be a Scott-continuous function such that $F \circ G = id_{\mathcal{C}}$. Then $\mathcal{C}$ inherits the structure of an algebraic domain.*

*Proof.* The proof is divided into two parts: first we show that $\mathcal{C}$ is a CPO, and then that it is algebraic.

### 6.0.1 Part 1: $\mathcal{C}$ is a CPO

1. **Minimum element**: Let $\bot_{\overline{\mathcal{T}}}$ be the minimum of $\overline{\mathcal{T}}$. Since $F$ is surjective and Scott-continuous, we have that $F(\bot_{\overline{\mathcal{T}}})$ is the minimum of $\mathcal{C}$. For any $c \in \mathcal{C}$, there exists $t \in \overline{\mathcal{T}}$ with $F(t) = c$. Since $\bot_{\overline{\mathcal{T}}} \leq t$, by monotonicity of $F$ (due to Scott-continuity), $F(\bot_{\overline{\mathcal{T}}}) \leq F(t) = c$.
2. **Directed suprema**: Let $D \subseteq \mathcal{C}$ be a directed set. Since $G$ is Scott-continuous, $G(D)$ is a directed set in $\overline{\mathcal{T}}$ (because $G$ preserves order). Let $s = \bigsqcup G(D)$ in $\overline{\mathcal{T}}$. Then we define $\bigsqcup D = F(s)$. To verify that this is the supremum of $D$:
    - For any $d \in D$, $G(d) \leq s$, so $d = F(G(d)) \leq F(s)$ by monotonicity of $F$.
    - If $c$ is an upper bound of $D$, then $G(c)$ is an upper bound of $G(D)$, so $s \leq G(c)$ and therefore $F(s) \leq F(G(c)) = c$.

### 6.0.2 Part 2: $\mathcal{C}$ is algebraic

Let $K(\mathcal{C}) = \{F(k) : k \in K(\overline{\mathcal{T}})\}$. We show that these elements are compact in $\mathcal{C}$ and that every element of $\mathcal{C}$ is the supremum of the elements of $K(\mathcal{C})$ below it.

1. **Compactness of** $F(k)$: Let $k \in K(\overline{\mathcal{T}})$ and let $D \subseteq \mathcal{C}$ be a directed set with $F(k) \leq \bigsqcup D$. Then:

   $$k \leq G(F(k)) \leq G\left(\bigsqcup D\right) = \bigsqcup G(D)$$

   where the last equality is due to the Scott-continuity of $G$. Since $k$ is compact in $\overline{\mathcal{T}}$, there exists $d \in D$ such that $k \leq G(d)$. Then, $F(k) \leq F(G(d)) = d$. Therefore, $F(k)$ is compact in $\mathcal{C}$.

2. **Approximation by compact elements**: For any $c \in \mathcal{C}$, we have $c = F(G(c))$. Since $\overline{\mathcal{T}}$ is algebraic, $G(c) = \bigsqcup\{k \in K(\overline{\mathcal{T}}) : k \leq G(c)\}$. By Scott-continuity of $F$:

   $$c = F(G(c)) = F\left(\bigsqcup\{k \in K(\overline{\mathcal{T}}) : k \leq G(c)\}\right) = \bigsqcup\{F(k) : k \in K(\overline{\mathcal{T}}), k \leq G(c)\}$$

   And for each $k \leq G(c)$, we have $F(k) \leq F(G(c)) = c$. Thus, $c$ is the supremum of compact elements in $\mathcal{C}$.

Therefore, $\mathcal{C}$ is an algebraic domain. □

## 7 Space of Scott-Continuous Functions

Our ultimate goal is to approximate a semantic function from $\overline{\mathcal{T}} \to \mathcal{C}$, so we define the space of continuous functions between posets:

**Definition 35** (Function Space)**.** *We define the space of Scott-continuous functions:*

$$[\overline{\mathcal{T}} \to \mathcal{C}] = \{f : \overline{\mathcal{T}} \to \mathcal{C} : f \textit{ is Scott-continuous}\}$$

*with the pointwise order:* $f \leq g \iff \forall t \in \overline{\mathcal{T}},\ f(t) \leq g(t)$.

**Remark 36.** *The class of Scott-continuous functions encompasses Lawson-continuous functions.*

**Remark 37** (Orders on $\mathcal{T}$)**.** *There is a trade-off between approximating a broader class of functions with a weaker order such as the prefix order, versus approximating with a finer topology using a stronger order such as the subsequence order.*

Functions between algebraic domains are completely characterized by their values on compact elements.

**Proposition 38** (Characterization of Functions on Algebraic Domains)**.** *If $D$ and $E$ are algebraic domains, then $f : D \to E$ is Scott-continuous if and only if:*

$$f(x) = \bigsqcup\{f(k) : k \in K(D),\ k \leq x\} \quad \textit{for every } x \in D$$

*Proof.* (⇒) By Scott-continuity and algebraicity:

$$f(x) = f\left(\bigsqcup\{k \in K(D) : k \leq x\}\right) = \bigsqcup\{f(k) : k \in K(D),\ k \leq x\}$$

(⇐) If $f$ is expressed as the supremum of its values on compact elements, then it preserves directed suprema. □

Since the compact elements in the completion $\overline{\mathcal{T}}$ are all finite texts in $\mathcal{T}$, we conclude something very important: defining a semantic function on $\mathcal{T}$ or on its completion $\overline{\mathcal{T}}$ is the same.

**Proposition 39.** *Let $f : \mathcal{T} \to \mathcal{C}$ be a Scott-continuous function. There exists a unique extension to its completion $\overline{f} : \overline{\mathcal{T}} \to \mathcal{C}$.*

On the other hand, since we are working with functions between two algebraic domains, it turns out that the function space inherits this good property.

**Theorem 40.** *Let $D, E$ be algebraic domains. The space $[D \to E]$ is an algebraic domain.*

Again, to define a notion of approximation, now for functions, we must give a topology to the function space. Taking advantage that it is itself a partially ordered set, we can define Lawson convergence.

Then, with this topology, it turns out that there exists a class of simple functions that are dense in the space.

## 7.1 Finite Step Functions

Finite step functions constitute the fundamental class of simple functions that will allow us to approximate any complex semantic function. Their finite and computable structure makes them ideal for modeling gradual language understanding processes.

**Definition 41** (Basic Step Function)**.** *Let $D$ and $E$ be algebraic domains. For each pair of compact elements $d \in K(D)$ and $e \in K(E)$, we define the* basic step function *$f_{d,e} : D \to E$ as:*

$$f_{d,e}(x) = \begin{cases} e & \textit{if } d \leq x \\ \bot & \textit{otherwise} \end{cases}$$

*where $\bot$ is the minimum element of $E$.*

**Example 42** (Basic Step Function in Text Space)**.** *Consider:*

- *$d =$ "cat" $\in K(\overline{\mathcal{T}})$ (compact text).*
- *$e =$ "feline" $\in K(\mathcal{C})$ (compact concept).*

*The basic step function $f_{\text{"cat"},\text{"feline"}}$ behaves as:*

$$f(x) = \begin{cases} \text{"feline"} & \textit{if the text } x \textit{ contains "cat" as a prefix} \\ \bot & \textit{otherwise} \end{cases}$$

***Evaluation examples:***

- $f($*"cat"*$) =$ *"feline"*.
- $f($*"black cat"*$) =$ *"feline"*.
- $f($*"dog"*$) = \perp$.
- $f($*"kitten"*$) = \perp$ *(not an exact prefix).*

**Definition 43** (Finite Step Function)**.** *A* finite step function *is the pointwise supremum of a finite set of basic step functions:*

$$g(x) = \bigsqcup \{f_{d_i,e_i}(x) : i = 1, \dots, n\} = \bigsqcup \{e_i \in K(\mathcal{C}) : d_i \leq x\}$$

*for some finite set* $\{(d_1, e_1), \dots, (d_n, e_n)\} \subseteq K(D) \times K(E)$*.*

**Remark 44.** *The function is well-defined because we saw that in $\mathcal{C}$ every finite subset has a supremum.*

**Remark 45.** *The cardinality of the image set is $|Im(f_{step})| < 2^n$, where $n$ is the number of basic step functions.*

These simple functions have the ability to approximate any Scott-continuous function, as stated below.

**Theorem 46** (Density of Finite Step Functions)**.** *The set of finite step functions is dense in $[D \to E]$ with the Lawson topology.*

The proof is left to the appendix theorem 50.

Finally, the mathematical structure we develop provides the foundations for the approximation theorem via Transformers, where finite step functions will be implemented exactly by specific neural architectures.

# 8 Universal Approximation Theorem for Semantic Functions via Transformers

We now proceed to state the main theorem of this section.

**Theorem 47** (Universal Approximation of Semantic Functions via Transformers)**.** *Let $F : \overline{\mathcal{T}} \to \mathcal{C}$ be a Scott-continuous function between algebraic domains. Then, for every Lawson neighborhood $U$ of $F$, there exists a Transformer architecture that implements a function contained in $U$.*

*Proof.* The proof proceeds in two fundamental steps:

Let $F \in [\overline{\mathcal{T}} \to \mathcal{C}]$.

**Step 1: Reduction to the case of step functions**

By the density theorem of compact functions (theorem 46), we know that the set of finite step functions is dense in the space $[\overline{\mathcal{T}} \to \mathcal{C}]$ with the Lawson topology.

Given $U$ a Lawson neighborhood of $F$, there exists a finite step function $g \in U$. Therefore, it suffices to prove that every finite step function can be implemented exactly by a Transformer architecture.

**Step 2: Exact implementation of finite step functions**

Let $g : \overline{\mathcal{T}} \to \mathcal{C}$ be a finite step function. It has a representation of the form:

$$g(x) = \bigsqcup \{f_{d_i,e_i}(x) : i = 1, \dots, n\} = \bigsqcup \{e_i \in K(\mathcal{C}) : d_i \leq x\}$$

for some finite set $\{(d_1, e_1), \dots, (d_n, e_n)\} \subseteq K(D) \times K(E)$.

We construct a Transformer that implements exactly $g$ through the following components:

1. **Prefix detection**: For each compact element $d_i$ that appears in the basic step functions, we implement an attention head that computes exactly the indicator $\mathbf{1}_{\{d_i \leq x\}}$.
2. **Supremum computation**: We implement a FFN that computes exactly the supremum $\bigsqcup \{e_i \in K(\mathcal{C}) : d_i \leq x\}$.

The sequential composition of these modules produces a Transformer that computes exactly $g(x)$. Since $g \in U$, we have demonstrated the existence of the required Transformer. □

**Lemma 48** (Exact Prefix Detection with a Single Attention Head)**.** *For any prefix $p = (p_1, \dots, p_m) \in K(\overline{\mathcal{T}})$, there exists a configuration of a single attention head and a FFN that computes exactly $\mathbf{1}\{p \leq x\}$.*

*Proof.* We use a positional one-hot encoding, where each word $w_i$ at position $i$ is encoded with a token $x_i$ represented as a vector in $\{0,1\}^{|V|+L_{\max}}$, with $|V|$ the vocabulary size and $L_{\max}$ the maximum length.

To detect if $p$ is a prefix of $w$, we need to verify that for each position $i = 1, \dots, m$, we have $w_i = p_i$. This is equivalent to computing the AND of the indicators $\mathbf{1}\{w_i = p_i\}$.

**Attention head configuration:**

We configure an attention head that simultaneously checks all positions of the prefix. Regarding the value matrix, we define it to indicate whether the word $w_i$ is correct.

- **Values**:

$$Vx_i = \begin{cases} 1 & \text{if } 1 \leq i \leq m \text{ and } w_i = p_i \\ 0 & \text{otherwise} \end{cases}$$

On the other hand, we configure the query $Q$ and key $K$ matrices so that for any word $x_i$ at position $i$:

- **Values**:

$$\langle Qx_i, Kx_i \rangle = \begin{cases} 1 & \text{if } 1 \leq i \leq m \\ 0 & \text{otherwise} \end{cases}$$

Note that then the output of the softmax will always be uniform among the first $m$ words, i.e., $\frac{1}{m}$.

Then the attention output will be:

$$a = \sum_{i=1}^{L} \text{softmax}(\langle Qx_i, Kx_i \rangle)) \cdot Vx_i = \sum_{i=1}^{m} \frac{1}{m} \cdot \mathbf{1}\{w_i = p_i\}$$

In the case where the prefix is present:

$$a = \sum_{j=1}^{m} \frac{1}{m} \cdot 1 = 1$$

In the case where at least one token of the prefix is missing:

$$a \leq \frac{m-1}{m} < 1$$

**Combination with FFN:**

The output $a$ is fed to a FFN with a threshold unit that computes:

$$\mathbf{1}\{p \leq x\} = \mathbf{1}\left\{a \geq 1 - \tfrac{1}{2m}\right\}$$

This condition is 1 if and only if $a = 1$, which occurs exactly when the complete prefix is present.

Therefore, the system composed of the single attention head and the FFN computes exactly $\mathbf{1}\{p \leq x\}$. □

**Lemma 49** (Exact Supremum Computation)**.** *The mapping* $\Phi : \mathcal{P}(\{c_1, .., c_n\}) \to \mathcal{C}$, $S \mapsto \bigsqcup S$ *can be represented exactly by a FFN.*

*Proof.* Since the domain and image are finite, we can construct the FFN component by component. Let $f$ be the one-hot encoding of the function that maps each subset to its supremum.

For each output position $j \in \{1, \ldots, n\}$, we define the component function $f_j : \{0,1\}^n \to \{0,1\}$ where $f_j(x) = [f(x)]_j$.

**Construction for each component $f_j$:**

For each input vector $a \in \{0,1\}^n$, we define a specific activation pattern. Since there are $2^n$ possible input vectors, let's enumerate them as $a^1, a^2, \ldots, a^{2^n}$.

For each pattern $a^k$, we define a hidden neuron $h_k$ that activates exactly when the input is $a^k$:

$$h_k(x) = \mathbf{1}\left\{\sum_{i=1}^{n} (2a_i^k - 1)x_i \geq n - \tfrac{1}{2}\right\}$$

The weights $(2a_i^k - 1)$ ensure that: - If $a_i^k = 1$, it contributes $+1$. - If $a_i^k = 0$, it contributes $-1$.

The sum $\sum_{i=1}^{n}(2a_i^k - 1)x_i$ reaches the maximum value $n$ only when $x_i = a_i^k$ for all $i$.

**Output layer:**

For each component $j$ of the output, we define:

$$y_j = \mathbf{1}\left\{\sum_{k=1}^{2^n} f_j(a^k) \cdot h_k(x) \geq \tfrac{1}{2}\right\}$$

Since the $h_k$ are mutually exclusive (exactly one is 1 for each input $x$), we have:

$$y_j = f_j(a^k) \quad \text{when } h_k(x) = 1$$

That is, $y_j = f_j(x)$ for every input $x$.

Therefore, this FFN computes exactly $f(x)$ for all $x \in \{0,1\}^n$. □

# 9 Extensions and Future Work

The theoretical framework developed suggests several promising directions:

- **Specialized algorithms**: Design NLP methods that explicitly exploit the partial order structure in conceptual spaces, potentially improving efficiency and interpretability, with possible significant impact on model reasoning capabilities.
- **Comparison of current embedding space with the theoretical one**: Study and compare the representation space of language models against this ideal ordered representation, using topological invariants.

# Acknowledgments

We thank the anonymous reviewers for their helpful comments.

# A Proof of Theorem 46

**Theorem 50** (Density of Finite Step Functions)**.** *The set of finite step functions is dense in $[D \to E]$ with the Lawson topology.*

*Proof.* Let $f \in [D \to E]$ be a continuous function and let $V$ be an open neighborhood of $f$ in the Lawson topology. We may assume that $V$ is a basic set of the form:

$$V = U \cap ([D \to E] \setminus \downarrow F)$$

where $U$ is a Scott open set with $f \in U$ and $F$ is a finite set with $f \notin \downarrow F$ (i.e., $f \not\leq h$ for every $h \in F$).

For each $h \in F$, since $f \not\leq h$, there exists $x_h \in D$ such that $f(x_h) \not\leq h(x_h)$. Since $E$ is algebraic, $f(x_h)$ is the directed supremum of compact elements below it, so there exists $e_h \in K(E)$ with $e_h \leq f(x_h)$ but $e_h \not\leq h(x_h)$.

By continuity of $f$, the set $\{x \in D : e_h \leq f(x)\}$ is a Scott open set in $D$ (because $e_h$ is compact) and it contains $x_h$. Since $D$ is algebraic, there exists $d_h \in K(D)$ with $d_h \leq x_h$ such that $e_h \leq f(d_h)$.

For each $h \in F$, consider the basic step function $f_{d_h,e_h}$. Define the finite step function:

$$g = \sup\{f_{d_h,e_h} : h \in F\}.$$

Since $F$ is finite, $g$ is a finite step function.

We claim that $g \leq f$. Indeed, for each $h \in F$ and $x \in D$:

- If $d_h \leq x$, then $f_{d_h,e_h}(x) = e_h \leq f(d_h) \leq f(x)$ by monotonicity of $f$.
- If $d_h \not\leq x$, then $f_{d_h,e_h}(x) = \perp \leq f(x)$.

Therefore, $f_{d_h,e_h} \leq f$ for each $h \in F$, and consequently $g \leq f$.

Since $f \in U$ and $U$ is a Scott open set (which is a lower set), it follows that $g \in U$.

Now we verify that $g \notin \downarrow F$. Suppose by contradiction that $g \in \downarrow F$, i.e., there exists $h' \in F$ such that $g \leq h'$. Consider $d_{h'} \in K(D)$. Since $d_{h'} \leq d_{h'}$, we have:

$$g(d_{h'}) \geq f_{d_{h'},e_{h'}}(d_{h'}) = e_{h'}.$$

But $g \leq h'$ implies $g(d_{h'}) \leq h'(d_{h'})$, so $e_{h'} \leq h'(d_{h'})$, which contradicts $e_{h'} \not\leq h'(d_{h'})$. Hence, $g \notin \downarrow F$.

Consequently, $g \in U \cap ([D \to E] \setminus \downarrow F) = V$. □

## References

[1] Samson Abramsky and Achim Jung. Domain theory. In *Handbook of Logic in Computer Science, Volume 3*. Oxford University Press, 1994.

[2] Jon Barwise and John Perry. *Situations and Attitudes*. MIT Press, 1983.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

[4] Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.

[5] Gerhard Gierz, Karl H. Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael Mislove, and Dana S. Scott. *Continuous Lattices and Domains*. Cambridge University Press, 2003.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[7] Richard Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, 1974.

[8] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[9] Gordon D. Plotkin. Domains. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B*. Elsevier, 1990.

[10] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations (ICLR)*, 2020.