Learning High-Order Word Representations

Konstantinos Kogkalidis

June 12, 2018

Logic & Language Fan Club

Motivation

Categorical Compositional Distributional Semantics

Idea: structure-preserving map $\mathcal{F}$

$$\mathcal{F} : \mathcal{G} \to \textit{\textbf{FdVect}}$$

- Atomic types translated to vectors (order-one tensors)
- Complex types translated to (multi-)linear maps (higher order tensors)

Why Compositionality?

- Bridging of formal & distributional semantics
- Syntax-informed meaning derivations
- Modeling of functional words
- Formal treatment of ambiguous words
- Richer representations

  ⋮

Why Not Compositionality?

- ✓ Great properties
- ? How to obtain word representations?

Possible options:

1. Co-occurrence statistics

Why Not Compositionality?

- ✓ Great properties
- ? How to obtain word representations?

Possible options:

1. Co-occurrence statistics ✗

Why Not Compositionality?

- ✓ Great properties
- **?** How to obtain word representations?

Possible options:

1. Co-occurrence statistics ✗
2. Unsupervised techniques (*a la word2vec*)

Why Not Compositionality?

- ✓ Great properties
- **?** How to obtain word representations?

Possible options:

1. Co-occurrence statistics ✗
2. Unsupervised techniques (*a la word2vec*) ✗

Why Not Compositionality?

✓ Great properties

? How to obtain word representations?

Possible options:

1. Co-occurrence statistics ✗
2. Unsupervised techniques (*a la word2vec*) ✗
3. Supervised learning **?**

Supervised Learning

Functional Overview

- Search over set of functions $A \to B$ parameterized over P
- Find optimal approximation $\hat{f}_P$ to $f : A \to B$
- Use samples $(a, f(a)) \in A \times B$ to update P

Supervised Learning

Dataset

Finding Data

Sample space must be:

- Labeled
- constrained
- of large size
- of high quality

Paraphrase Database

Raw text paraphrase pairs

Example pair

proposed by the president $\sim$ *suggested by the chairman*

- Labeled ✓
- constrained ✗ *(different syntactic types)*
- of large size ✓
- of high quality **?**

Dataset: Preprocessing

1. **Parse and filter by type**

- Labeled ✓
- constrained ✓
- of large size ✗ *(>95% loss)*
- of high quality ✗ *(parser-induced errors)*

Dataset: Preprocessing

2. **Back-translation**

- Labeled ✓
- constrained ✓
- of large size ✓
- of high quality ✗ *(translation-induced errors)*

Dataset: Preprocessing

3. **Filter by co-occurrence / mutual information**

- Labeled ✓
- constrained ✓
- of large size ✓
- of high quality **?**

Dataset: End Product

Verb / object dictionaries:

$$\mathcal{V} : \{v_1 : 1, v_2 : 2, ..., v_N : N\}$$
$$\mathcal{O} : \{o_1 : 1, o_2 : 2, ..., o_M : M\}$$

Paraphrase relation:

$$\mathcal{P} : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \{0, 1\} \qquad \text{(binary classification)}$$
$$\mathcal{P}(i, j, k, l) = \mathcal{P}(k, l, i, j) = \begin{cases} 1 & v_i o_j \sim v_k o_l \\ 0 & \text{otherwise} \end{cases}$$

Supervised Learning

Intermediate Representations

Objective Function

Our semantic interpretations are:

- Sentences: $\lceil s \rceil = \mathbb{R}^S$
- Objects: $\lceil np \rceil = \mathbb{R}^{NP}$
- Verbs: $\lceil s/np \rceil = \mathbb{R}^{S \times NP}$

And our objective is to learn a verb embedding function ε_{verb}:

$$\varepsilon_{verb} : \mathbb{N} \to \mathbb{R}^{S \times NP}$$

But instead we have samples from some $f : \mathbb{N}^4 \to \{0, 1\}$

Formulating the network

Solution

Formulate f_P to incorporate ε_{verb}.

$$f_p = f_1 \circ f_2 \circ \cdots \circ \varepsilon_{verb} \circ \ldots$$

Formulating the network

Solution

Formulate f_P to incorporate ε_{verb}.

$$f_p = f_1 \circ f_2 \circ \cdots \circ \varepsilon_{verb} \circ \ldots$$

Simplification

Assume pre-trained object embedding function ε_{object}

$$\varepsilon_{objects} : \mathbb{N} \to \mathbb{R}^{300}$$

Filling the missing blocks

$$
\begin{array}{cccc}
i \in \mathbb{N} & j \in \mathbb{N} & k \in \mathbb{N} & l \in \mathbb{N} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
i \in \mathbb{R}^{100\times 300} & j \in \mathbb{R}^{300} & k \in \mathbb{R}^{100\times 300} & l \in \mathbb{R}^{300}
\end{array}
$$