Self-Adaptive Hierarchical Sentence Model

Han Zhao, Zhengdong Lu and Pascal Poupart

UNIVERSITY OF WATERLOO

han.zhao@uwaterloo.ca

July 22, 2015
Outline

Background
  Machine Learning
  Representation Learning

AdaSent
  Motivation
  Architecture
  Learning
  Experiments
Definition
A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.

— Tom M. Mitchell
Definition
A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.

— Tom M. Mitchell

$E$ – data
Definition
A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.

— Tom M. Mitchell

- $E$ – data
- $T$ – task of interests
Definition
A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.

— Tom M. Mitchell

- $E$ – data
- $T$ – task of interests
- $P$ – objective function
Background

Machine Learning – Application
Background

Machine Learning – Pipeline

Slide courtesy of Kyunghyun Cho
Background
Machine Learning – Components

Machine Learning ≈ Representation + Objective + Optimization
Background

Machine Learning – Components

Machine Learning ≈ **Representation** + Objective + Optimization
Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.
Background

Deep Learning – Application

- **Object Recognition** (Krizhevsky et al. 2012)
- Speech Recognition (Graves et al. 2013)
- **Neural Machine Translation** (Sutskever et al. 2014)
- Face Recognition (Schroff et al. 2015)
- Deep Reinforcement Learning (Mnih et al. 2013)
- Image Caption Generation (Vinyals et al. 2014)
- Text Matching (Hu et al. 2014)
- Text Parsing (Chen et al. 2014)

And etc.

Items highlighted in blue happen at Google.
Background
Deep Learning – Representation Learning

**Deep Learning:** Learning multiple levels of representation directly from massive data
Background
Deep Learning – Representation Learning

Deep Learning: Learning multiple levels of representation directly from massive data

\[ a_i^l = \sigma(z_i^l), \quad z_i^l = \sum_j w_{ij} a_j^{l-1}, \quad \sigma(t) = \frac{1}{1 + \exp^{-t}} \]
Background

Deep Learning – Natural Language Processing

word2vec
Tool for computing continuous distributed representations of words.
Background
Deep Learning – Natural Language Processing

**word2vec**
Tool for computing continuous distributed representations of words.
Traditional representation of words: One-hot representation.

\[
\begin{align*}
\text{cat} & = [0, 0, 0, 0, 0, 1, 0, 0, \ldots] \\
\text{dog} & = [0, 1, 0, 0, 0, 0, 0, 0, \ldots]
\end{align*}
\]
Traditional representation of words: One-hot representation.

\[
\begin{align*}
\text{cat} &= [0, 0, 0, 0, 0, 1, 0, 0, \ldots] \\
\text{dog} &= [0, 1, 0, 0, 0, 0, 0, 0, \ldots]
\end{align*}
\]

**Pros:**
- Simple, intuitive
- Basis of bag-of-words model for document representation

**Cons:**
- High-dimensional
- No semantic meaning
Background
Deep Learning – Word Embedding

**Word2Vec (Minkolov et al)**

Distributed word representation: Unsupervised technique to map each word into a dense and real-valued low dimensional vector.
Background
Deep Learning – Word Embedding

Word2Vec (Minkolov et al)

Distributed word representation: Unsupervised technique to map each word into a dense and real-valued low dimensional vector.

\[ w(\text{China}) - w(\text{Beijing}) \approx w(\text{Russia}) - w(\text{Moscow}) \approx w(\text{Italy}) - w(\text{Rome}) \]
Background
Deep Learning – Sentence Modeling

Paragraph Vector (Le et al)
Distributed paragraph representation: Unsupervised technique to map each paragraph into a dense and real-valued low dimensional vector.
Background
Deep Learning – Sentence Modeling

Phrase/Sentence/Document Modeling

- Recursive auto-encoder/Matrix-vector recursive neural network (Socher et al. 2011, 2012)
- Convolutional neural network (Kim 2014)
- Dynamic convolutional neural network (Kalchbrenner et al. 2014)
- Recurrent neural network/Bi-directional recurrent neural network (Lai et al. 2015)
- Gated recursive convolutional neural network (Cho et al. 2014)
AdaSent: Self-Adaptive Hierarchical Sentence Model

- Is vector representation with fixed-length enough to represent different granularities of phrases/sentences/documents?
AdaSent
Motivation

AdaSent: Self-Adaptive Hierarchical Sentence Model

- Is vector representation with fixed-length enough to represent different granularities of phrases/sentences/documents?
- Can we model the composition behaviour using algebraic operations with enough flexibility?
AdaSent
Motivation

AdaSent: Self-Adaptive Hierarchical Sentence Model

- Is vector representation with fixed-length enough to represent different granularities of phrases/sentences/documents?
- Can we model the composition behaviour using algebraic operations with enough flexibility?
- Can we design a model which can decide the representation of phrases/sentences on the fly based on the current task at hand?
AdaSent

Architecture

Three components:
Three components:

- Composition hierarchy
AdaSent

Architecture

Three components:
- Composition hierarchy
- Gating network
AdaSent

Architecture

Three components:

- Composition hierarchy
- Gating network
- Classifier
Properties of AdaSent

- Maintains a hierarchy of abstractions from the raw input, rather than a fixed length vector representation.
- Implements $N$-gram model where $N$ ranges from 1 to the length of the sentence.
- Implements and extends the mixture-of-experts idea.
- Final decision is based on an ensemble of different level of abstractions.
AdaSent
Architecture

Composition Pyramid

Directed acyclic graph whose height depends on the length of input sentence.

Composition dynamics:

\[
\begin{align*}
    h_j^t &= \omega_l h_{j-1}^t + \omega_r h_{j+1}^t + \omega_c \tilde{h}_j^t \\
    \tilde{h}_j^t &= f(W_L h_j^{t-1} + W_R h_{j+1}^{t-1} + b_W)
\end{align*}
\]

Local combination parametrizations:

\[
\begin{pmatrix}
    \omega_l \\
    \omega_r \\
    \omega_c
\end{pmatrix} = \text{softmax} \left( G_L h_j^{t-1} + G_R h_{j+1}^{t-1} + b_G \right)
\]

where \( W_L, W_R \in \mathbb{R}^{D \times D} \) and \( G_L, G_R \in \mathbb{R}^{3 \times D} \).
AdaSent
Architecture

Composition Pyramid

Intuitive interpretation:

$$= \omega_i^2 \left( \omega_{i1} + \omega_{r1} + \omega_{c1} \right)$$

$$+ \omega_r^2 \left( \omega_{i2} + \omega_{r2} + \omega_{c2} \right)$$

$$+ \omega_c^2$$
Level Pooling

Global (average/max) pooling applied to each level of the pyramid to build the abstraction in the hierarchy.

Average pooling:

$$\bar{h} = \frac{1}{T} \sum_{t=1}^{T} h_t$$

Max pooling:

$$\bar{h}_j = \max_{t \in 1:T} h_{tj}, \quad \forall j \in 1:D$$
Gating Network and Classifier

Gating network: $\omega : \mathbb{R}^D \mapsto \mathbb{R}_+$. Let $\gamma_t \triangleq \omega(\bar{h}_t)$. Constraint: $\sum_{t=1}^{T} \omega(\bar{h}_t) = 1$. Let $g : \mathbb{R}^D \mapsto \Delta_+$ be the classification function.

Classification consensus

$$p(C = c|x_1:T) = \sum_{t=1}^{T} p(c|\mathcal{H}_x = t) \cdot p(\mathcal{H}_x = t|x) = \sum_{t=1}^{T} g_c(\bar{h}_t) \cdot \omega(\bar{h}_t)$$
Backpropagation through Structure (BPTS)

Partial derivative of objective function $L$ with respect to model parameters:

\[
\frac{\partial L}{\partial W_L} = \sum_{t=1}^{T} \sum_{j=1}^{T-t+1} \frac{\partial L}{\partial h_j^t} \frac{\partial h_j^t}{\partial W_L}, \quad \frac{\partial L}{\partial W_R} = \sum_{t=1}^{T} \sum_{j=1}^{T-t+1} \frac{\partial L}{\partial h_j^t} \frac{\partial h_j^t}{\partial W_R}
\]

where

\[
\frac{\partial L}{\partial h_j^t} = \frac{\partial L}{\partial h_j^{t+1}} \frac{\partial h_j^{t+1}}{\partial h_j^t} + \frac{\partial L}{\partial h_j^{t+1}} \frac{\partial h_j^{t+1}}{\partial h_j^{t-1}}
\]

\[
\frac{\partial h_j^{t+1}}{\partial h_j^t} = \omega_r I + \omega_c \text{diag}(f') W_R, \quad \frac{\partial h_j^{t+1}}{\partial h_j^t} = \omega_l I + \omega_c \text{diag}(f') W_L
\]
AdaSent
Experiments

Data Sets

- **MR.** Movie reviews data set where each instance is a sentence. The objective is to classify each review by its overall sentiment polarity, either positive or negative.

- **CR.** Annotated customer reviews of 14 products obtained from Amazon. The task is to classify each customer review into positive and negative categories.

- **SUBJ.** Subjectivity data set where the goal is to classify each instance (snippet) as being subjective or objective.

- **MPQA.** Phrase level opinion polarity detection subtask of the MPQA data set.

- **TREC.** Question data set, in which the goal is to classify an instance (question) into 6 different types.
## Data Sets

| Data | $N$   | $\text{dist}(+,-)$            | $K$ | $|w|$ | test |
|------|-------|-------------------------------|-----|------|------|
| MR   | 10662 | (0.5, 0.5)                    | 2   | 18   | CV   |
| CR   | 3788  | (0.64, 0.36)                  | 2   | 17   | CV   |
| SUBJ | 10000 | (0.5, 0.5)                    | 2   | 21   | CV   |
| MPQA | 10099 | (0.31, 0.69)                  | 2   | 3    | CV   |
| TREC | 5952  | (0.1,0.2,0.2,0.1,0.2,0.2)     | 6   | 10   | 500  |

**Table:** $N$ counts the number of instances and $\text{dist}$ lists the class distribution in the data set. $K$ represents the number of target classes. $|w|$ measures the average number of words in each instance. **test** is the size of the test set.
## AdaSent

### Experiments

### Classification Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>MR</th>
<th>CR</th>
<th>SUBJ</th>
<th>MPQA</th>
<th>TREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB-SVM</td>
<td>79.4</td>
<td>81.8</td>
<td>93.2</td>
<td>86.3</td>
<td>-</td>
</tr>
<tr>
<td>MNB</td>
<td>79.0</td>
<td>80.0</td>
<td>93.6</td>
<td>86.3</td>
<td>-</td>
</tr>
<tr>
<td>RAE</td>
<td>77.7</td>
<td>-</td>
<td>-</td>
<td>86.4</td>
<td>-</td>
</tr>
<tr>
<td>MV-RecNN</td>
<td>79.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CNN</td>
<td>81.5</td>
<td>85.0</td>
<td>93.4</td>
<td>89.6</td>
<td><strong>93.6</strong></td>
</tr>
<tr>
<td>DCNN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>93.0</td>
</tr>
<tr>
<td>P.V.</td>
<td>74.8</td>
<td>78.1</td>
<td>90.5</td>
<td>74.2</td>
<td>91.8</td>
</tr>
<tr>
<td>cBoW</td>
<td>77.2</td>
<td>79.9</td>
<td>91.3</td>
<td>86.4</td>
<td>87.3</td>
</tr>
<tr>
<td>RNN</td>
<td>77.2</td>
<td>82.3</td>
<td>93.7</td>
<td>90.1</td>
<td>90.2</td>
</tr>
<tr>
<td>BRNN</td>
<td>82.3</td>
<td>82.6</td>
<td>94.2</td>
<td>90.3</td>
<td>91.0</td>
</tr>
<tr>
<td>GrConv</td>
<td>76.3</td>
<td>81.3</td>
<td>89.5</td>
<td>84.5</td>
<td>88.4</td>
</tr>
<tr>
<td>AdaSent</td>
<td><strong>83.1</strong></td>
<td><strong>86.3</strong></td>
<td><strong>95.5</strong></td>
<td><strong>93.3</strong></td>
<td>92.4</td>
</tr>
</tbody>
</table>
## AdaSent

### Experiments

#### Model Variance

<table>
<thead>
<tr>
<th>Model</th>
<th>MR</th>
<th>CR</th>
<th>SUBJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.V.</td>
<td>71.11 ± 0.80</td>
<td>71.22 ± 1.04</td>
<td>90.22 ± 0.21</td>
</tr>
<tr>
<td>cBoW</td>
<td>72.74 ± 1.03</td>
<td>71.86 ± 2.00</td>
<td>90.58 ± 0.52</td>
</tr>
<tr>
<td>RNN</td>
<td>74.39 ± 1.70</td>
<td>73.81 ± 3.52</td>
<td>89.97 ± 2.88</td>
</tr>
<tr>
<td>BRNN</td>
<td>75.25 ± 1.33</td>
<td>76.72 ± 2.78</td>
<td>90.93 ± 1.00</td>
</tr>
<tr>
<td>GrConv</td>
<td>71.64 ± 2.09</td>
<td>71.52 ± 4.18</td>
<td>86.53 ± 1.33</td>
</tr>
<tr>
<td>AdaSent</td>
<td><strong>79.84 ± 1.26</strong></td>
<td><strong>83.61 ± 1.60</strong></td>
<td><strong>92.19 ± 1.19</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>MPQA</th>
<th>TREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.V.</td>
<td>67.93 ± 0.57</td>
<td>86.30 ± 1.10</td>
</tr>
<tr>
<td>cBoW</td>
<td>84.04 ± 1.20</td>
<td>85.16 ± 1.76</td>
</tr>
<tr>
<td>RNN</td>
<td>84.52 ± 1.17</td>
<td>84.24 ± 2.61</td>
</tr>
<tr>
<td>BRNN</td>
<td>85.36 ± 1.13</td>
<td>86.28 ± 0.90</td>
</tr>
<tr>
<td>GrConv</td>
<td>82.00 ± 0.88</td>
<td>82.04 ± 2.23</td>
</tr>
<tr>
<td>AdaSent</td>
<td><strong>90.42 ± 0.71</strong></td>
<td><strong>91.10 ± 1.04</strong></td>
</tr>
</tbody>
</table>
Figure: Each row corresponds to the belief score of a sentence of length 12 sampled from one of the data sets. From top to bottom, the 10 sentences are sampled from MR, CR, SUBJ, MPQA and TREC respectively.
Concrete Example

Sentence: If the movie were all comedy it might work better but it has an ambition to say something about its subjects but not willingness.
AdaSent
Experiments

Representation Learning - SUBJ

Figure: AdaSent

Figure: Original
AdaSent
Experiments

Representation Learning - MPQA

Figure: AdaSent

Figure: Original
AdaSent
Experiments

Representation Learning - TREC

Figure: AdaSent

Figure: Original
Thanks

Question and Answering


International Joint Conference on Artificial Intelligence 2015