Deep Unordered Composition Rivals Syntactic Methods for Text Classification

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III

University of Maryland, College Park
University of Colorado, Boulder
Vector Space Models for NLP

- Represent words by low-dimensional vectors called **embeddings**
From One Word to Many Words

• How do we **compose** word embeddings into vectors that capture the meanings of phrases, sentences, and documents?
From One Word to Many Words

• How do we *compose* word embeddings into vectors that capture the meanings of phrases, sentences, and documents?

I  love  their  music
From One Word to Many Words

• How do we \textbf{compose} word embeddings into vectors that capture the meanings of phrases, sentences, and documents?

\[ g( I \text{ love their music} ) = \]
Task-Specific Composition Functions

• Sentiment Analysis

• Factoid Question Answering

• Machine Translation

• Parsing

• Image Captioning

• Generation

• Lots more!
Task-Specific Composition Functions

- Sentiment Analysis
- Factoid Question Answering
- Machine Translation
- Parsing
- Image Captioning
- Generation
- Lots more!

Our main contribution: A fast and simple composition function that competes with more complex methods on these two tasks.
Outline

• Review of composition functions
• Deep averaging networks (DAN)
• Experiments (factoid QA & sentiment analysis)
• How do DANs work?
• Error analysis & comparisons to previous work
Two Types of Composition

1. Unordered (bag-of-words)

\[ g( l \ love \ their \ music ) = \]
Two Types of Composition

1. Unordered (bag-of-words)

\[ g(\text{love, music, I, their}) = \]
Two Types of Composition

1. Unordered (bag-of-words)

\[ g(\text{love music, l, their}) = \] 

2. Syntactic (incorporates word order and syntax)

\[ g(\text{l, love, their, music}) = \]
Two Types of Composition

1. Unordered (bag-of-words)

\[ g(\text{love music}, \text{l}, \text{their}) = \] 

2. Syntactic (incorporates word order and syntax)

\[ g(\text{l}, \text{love}, \text{their}, \text{music}) = \]
Unordered Composition: the **NBO**W

- Apply a simple element-wise vector operation to all word embeddings; a **neural bag-of-words**
  - e.g., addition, multiplication, averaging
- Advantages: very fast, simple to implement
- Used previously as a baseline model (e.g., Kalchbrenner & Blunsom, 2014)
An **NBO\textbf{W}** for Sentiment Analysis
An **NBOW** for Sentiment Analysis

**Predator** is a **masterpiece**
An **NBOW** for Sentiment Analysis

\[ av = \sum_{i=1}^{4} \frac{c_i}{4} \]

Predator is a masterpiece

\[ c_1 \quad c_2 \quad c_3 \quad c_4 \]
An **NBOW** for Sentiment Analysis

softmax: predict positive label

$$a v = \sum_{i=1}^{4} \frac{c_i}{4}$$

Predator is a masterpiece

$C_1$ $C_2$ $C_3$ $C_4$
An **NBOW** for Sentiment Analysis

softmax: predict positive label

\[ av = \sum_{i=1}^{4} \frac{c_i}{4} \]

Predator is a masterpiece

Relatively low performance on classification tasks!
Syntactic Composition

- Neural network-based approaches
  - Recursive
  - Recurrent
  - Convolutional

- Advantages: usually yield higher accuracies than unordered functions on downstream tasks
Syntactic Composition

• Neural network-based approaches
  • Recursive
  • Recurrent
  • Convolutional

• Advantages: usually yield higher accuracies than unordered functions on downstream tasks
Recursive Neural Networks (RecNN)

• $g$ depends on a parse tree of the input text sequence
Recursive Neural Networks (RecNN)

• \( g \) depends on a parse tree of the input text sequence

Predator is a masterpiece

\[ c_1 \quad c_2 \quad c_3 \quad c_4 \]
Recursive Neural Networks (**RecNN**)

- $g$ depends on a *parse tree* of the input text sequence

![Diagram of a parse tree with nodes labeled 'Predator', 'is', 'a', 'masterpiece'.](image)

$$z_1 = f(W \begin{bmatrix} c_3 \\ c_4 \end{bmatrix})$$
Recursive Neural Networks (RecNN)

- \( g \) depends on a *parse tree* of the input text sequence.

\[
z_1 = f(W \begin{bmatrix} c_3 \\ c_4 \end{bmatrix})
\]

\[
z_2 = f(W \begin{bmatrix} c_2 \\ z_1 \end{bmatrix})
\]

Predator is a masterpiece

- \( c_1 \)
- \( c_2 \)
- \( c_3 \)
- \( c_4 \)
Recursive Neural Networks (RecNN)

- $g$ depends on a parse tree of the input text sequence

$z_3 = f(W \begin{bmatrix} c_1 \\ z_2 \end{bmatrix})$

$z_2 = f(W \begin{bmatrix} c_2 \\ z_1 \end{bmatrix})$

$z_1 = f(W \begin{bmatrix} c_3 \\ c_4 \end{bmatrix})$

Predator is a masterpiece

$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$
Recursive Neural Networks (RecNN)

- \( \mathbf{g} \) depends on a parse tree of the input text sequence.
Isolating the Impact of Syntax

- **RecNNs** have two advantages over **NBOW** models: syntax (obviously) and *nonlinear transformations*

- removing nonlinearities from **RecNNs** decreases absolute sentiment classification accuracy by over 5% (Socher et al., 2013)

- **NBOWs** are linear mappings between embeddings and outputs… what happens if we add nonlinearities?
Deep Averaging Networks

\[ av = \sum_{i=1}^{4} \frac{c_i}{4} \]

Predator is a masterpiece

C_1 \quad C_2 \quad C_3 \quad C_4
Deep Averaging Networks

\[ z_1 = f(W_1 \cdot \text{av}) \]

\[ \text{av} = \sum_{i=1}^{4} \frac{c_i}{4} \]

Predator is a masterpiece

\( c_1 \)  \( c_2 \)  \( c_3 \)  \( c_4 \)
Deep Averaging Networks

\[ av = \sum_{i=1}^{4} \frac{c_i}{4} \]

\[ z_1 = f(W_1 \cdot av) \]

\[ z_2 = f(W_2 \cdot z_1) \]

Predator is a masterpiece

\( c_1 \) \( c_2 \) \( c_3 \) \( c_4 \)
Deep Averaging Networks

softmax: predict positive label

\[ z_2 = f(W_2 \cdot z_1) \]

\[ z_1 = f(W_1 \cdot \text{av}) \]

\[ \text{av} = \sum_{i=1}^{4} \frac{c_i}{4} \]

Predator is a masterpiece

\( c_1 \) \( c_2 \) \( c_3 \) \( c_4 \)
Experiments

Factoid Question Answering

Sentiment Analysis
QA: Quiz Bowl
This creature has female counterparts named Penny and Gown.
This creature has female counterparts named Penny and Gown.
This creature appears dressed in Viking armor and carrying an ax when he is used as the mascot of PaX, a least privilege protection patch.
This creature has female counterparts named Penny and Gown.

This creature appears dressed in Viking armor and carrying an ax when he is used as the mascot of PaX, a least privilege protection patch.

This creature’s counterparts include Daemon on the Berkeley Software Distribution, or BSD.
QA: Quiz Bowl

This creature has female counterparts named Penny and Gown.

This creature appears dressed in Viking armor and carrying an ax when he is used as the mascot of PaX, a least privilege protection patch.

This creature’s counterparts include Daemon on the Berkeley Software Distribution, or BSD.

For ten points, name this mascot of the Linux operating system, a penguin whose name refers to formal male attire.
This creature has female counterparts named Penny and Gown.

This creature appears dressed in Viking armor and carrying an ax when he is used as the mascot of PaX, a least privilege protection patch.

This creature’s counterparts include Daemon on the Berkeley Software Distribution, or BSD.

For ten points, name this mascot of the Linux operating system, a penguin whose name refers to formal male attire.

Answer: Tux
QA: Dataset

- Used in this work: history quiz bowl question dataset of Iyyer et al., 2014
  - original dataset: 3,761 question/answer pairs
  - +wiki dataset: original + 53,234 sentence/page-title pairs from Wikipedia
QA: Models

- **BoW-DT**: bag-of-unigrams logistic regression with dependency relations

- **IR**: an information retrieval system built with Whoosh, uses BM-25 term weighting, query expansion, and fuzzy query matching

- **QANTA**: a recursive neural network structured around dependency parse trees

- **DAN**: our model with three hidden layers, trained with word dropout regularization
# QA: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Pos 1</th>
<th>Pos 2</th>
<th>Full</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW-DT</td>
<td>35.4</td>
<td>57.7</td>
<td>60.2</td>
<td>—</td>
</tr>
<tr>
<td>IR</td>
<td>37.5</td>
<td>65.9</td>
<td>71.4</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA</td>
<td>47.1</td>
<td>72.1</td>
<td>73.7</td>
<td>314</td>
</tr>
<tr>
<td>DAN</td>
<td>46.4</td>
<td>70.8</td>
<td>71.8</td>
<td>18</td>
</tr>
</tbody>
</table>
## QA: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Pos 1</th>
<th>Pos 2</th>
<th>Full</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW-DT</td>
<td>35.4</td>
<td>57.7</td>
<td>60.2</td>
<td>—</td>
</tr>
<tr>
<td>IR</td>
<td>37.5</td>
<td>65.9</td>
<td>71.4</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA</td>
<td>47.1</td>
<td>72.1</td>
<td>73.7</td>
<td>314</td>
</tr>
<tr>
<td>DAN</td>
<td>46.4</td>
<td>70.8</td>
<td>71.8</td>
<td>18</td>
</tr>
<tr>
<td>IR-WIKI</td>
<td>53.7</td>
<td>76.6</td>
<td>77.5</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA-WIKI</td>
<td>46.5</td>
<td>72.8</td>
<td>73.9</td>
<td>1,648</td>
</tr>
<tr>
<td>DAN-WIKI</td>
<td><strong>54.8</strong></td>
<td>75.5</td>
<td>77.1</td>
<td>119</td>
</tr>
</tbody>
</table>
## QA: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Pos 1</th>
<th>Pos 2</th>
<th>Full</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW-DT</td>
<td>35.4</td>
<td>57.7</td>
<td>60.2</td>
<td>—</td>
</tr>
<tr>
<td>IR</td>
<td>37.5</td>
<td>65.9</td>
<td>71.4</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA</td>
<td>47.1</td>
<td>72.1</td>
<td>73.7</td>
<td>314</td>
</tr>
<tr>
<td>DAN</td>
<td>46.4</td>
<td>70.8</td>
<td>71.8</td>
<td>18</td>
</tr>
<tr>
<td>IR-WIKI</td>
<td>53.7</td>
<td>76.6</td>
<td>77.5</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA-WIKI</td>
<td>46.5</td>
<td>72.8</td>
<td>73.9</td>
<td>1,648</td>
</tr>
<tr>
<td>DAN-WIKI</td>
<td><strong>54.8</strong></td>
<td>75.5</td>
<td>77.1</td>
<td>119</td>
</tr>
</tbody>
</table>
DANs Handle Syntactic Diversity

• Sentences from Wikipedia are syntactically different from quiz bowl questions

QB: “Identify this British author who wrote Wuthering Heights” → very common imperative construction in QB

• They can also contain lots of noise!

WIKI: “She does not seem to have made any friends outside her family.” (from Emily Brontë’s page)
QA: Man vs. Machine

• Scaled up a **DAN** (in combination with language model features) to handle ~100k Q/A pairs with ~14k unique answers!

• Our system played a match against a team of four former multiple-day Jeopardy champions
QA: Man vs. Machine

• Scaled up a **DAN** (in combination with language model features) to handle ~100k Q/A pairs with ~14k unique answers!

• Our system played a match against a team of four former multiple-day Jeopardy champions

The result: a 200-200 **tie**!
QA: Man vs. Machine

• Scaled up a **DAN** (in combination with language model features) to handle ~100k Q/A pairs with ~14k unique answers!

• Our system played a match against a team of four former multiple-day Jeopardy champions

The result: a 200-200 **tie**!

Round 2 in October: our system duels Ken Jennings
Silly humans...
Sentiment: Datasets

• Sentence-level:
  • Rotten Tomatoes (RT) movie reviews (Pang & Lee, 2005): 5,331 positive and 5,331 negative sentences
  • Stanford Sentiment Treebank (SST) (Socher et al., 2013): modified version of RT with fine-grained phrase annotations

• Document-level:
  • IMDB movie review dataset (Maas et al., 2011): 12,500 positive reviews and 12,500 negative reviews
Sentiment: Syntactic Models

- Standard **RecNNs** and more powerful variants: deep **RecNN** (Irsoy & Cardie, 2014), **RecNTN** (Socher et al., 2013)

- Standard convolutional nets (**CNN-MC** of Kim, 2014) and **dynamic CNNs** (Kalchbrenner et al., 2014)

- Paragraph vector (Le & Mikolov, 2014), restricted Boltzmann machine (Dahl et al., 2012)
## Sentiment: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>RT</th>
<th>SST fine</th>
<th>SST binary</th>
<th>IMDB</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN</td>
<td>80.3</td>
<td>47.7</td>
<td>86.3</td>
<td>89.4</td>
<td>136</td>
</tr>
<tr>
<td>NBOw</td>
<td>79.0</td>
<td>43.6</td>
<td>83.6</td>
<td>89.0</td>
<td>91</td>
</tr>
</tbody>
</table>

- RT: Recall (Percentage)
- SST fine: Sentimental-Stereotype Task fine-grained
- SST binary: Sentimental-Stereotype Task binary
- IMDB: IMDb dataset
- Time (sec): Time taken in seconds
## Sentiment: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>RT</th>
<th>SST fine</th>
<th>SST binary</th>
<th>IMDB</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN</td>
<td>80.3</td>
<td>47.7</td>
<td>86.3</td>
<td>89.4</td>
<td>136</td>
</tr>
<tr>
<td>NBO</td>
<td>79.0</td>
<td>43.6</td>
<td>83.6</td>
<td>89.0</td>
<td>91</td>
</tr>
<tr>
<td>RecNN</td>
<td>77.7</td>
<td>43.2</td>
<td>82.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RecNTN</td>
<td>—</td>
<td>45.7</td>
<td>85.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DRecNN</td>
<td>—</td>
<td>49.8</td>
<td>86.6</td>
<td>—</td>
<td>431</td>
</tr>
<tr>
<td>TreeLSTM</td>
<td>—</td>
<td><strong>50.6</strong></td>
<td>86.9</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DCNN</td>
<td>—</td>
<td>48.5</td>
<td>86.9</td>
<td>89.4</td>
<td>—</td>
</tr>
<tr>
<td>PVEC</td>
<td>—</td>
<td>48.7</td>
<td>87.8</td>
<td><strong>92.6</strong></td>
<td>—</td>
</tr>
<tr>
<td>CNN-MC</td>
<td><strong>81.1</strong></td>
<td>47.4</td>
<td><strong>88.1</strong></td>
<td>—</td>
<td>2,452</td>
</tr>
<tr>
<td>WRRBM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>89.2</td>
<td>—</td>
</tr>
</tbody>
</table>
How do **DANs** work?
How do **DAN**s work?

• The film’s performances were **awesome**
What About Negations?

- We collect **48 positive** and **44 negative** sentences from the SST that each contain at least one negation and one contrastive conjunction.

- When confronted with a negation, both the unordered **DAN** and syntactic **DRecNN** predict negative sentiment around 70% of the time.

- Accuracy on only the **positive sentences** in our subset is low: 37.5% for the **DAN** and 41.7% for the **DRecNN**.
<table>
<thead>
<tr>
<th>Sentence</th>
<th>DAN</th>
<th>DRecNN</th>
<th>Ground-Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>blessed with immense physical prowess he may well be, but aholia is simply not an actor</td>
<td>positive</td>
<td>neutral</td>
<td>negative</td>
</tr>
<tr>
<td>too bad, but thanks to some lovely comedic moments and several fine performances, it's not a total loss</td>
<td>negative</td>
<td>negative</td>
<td>positive</td>
</tr>
<tr>
<td>it's so good that its relentless, polished wit can withstand not only inept school productions, but even oliver parker's movie adaptation</td>
<td>negative</td>
<td>positive</td>
<td>positive</td>
</tr>
<tr>
<td>the movie was bad</td>
<td>negative</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td>the movie was not bad</td>
<td>negative</td>
<td>negative</td>
<td>positive</td>
</tr>
</tbody>
</table>
Recap

• Introduced the **DAN** for fast and simple text classification

• Our findings suggest that non-linearly transforming input embeddings is crucial for performance

• Complex syntactic models make mistakes similar to those of the more naïve **DANs**... syntax is important, but we need more data and/or models that generalize with fewer examples
Thanks! Questions?

code@github.com/miyyer/dan