Dependency Link Embeddings: Continuous Representations of Syntactic Substructures

Mohit Bansal
TTI-Chicago
(n-ary word cluster/embedding features)

prefix6 $\rightarrow$ (110010, 000101)
prefix4 $\rightarrow$ (1100, 0001)

(Li et al., 2014; Chen and Manning, 2014)
Motivation

\[ w[i] \rightarrow (0.6, -0.9) \]

\[ \text{ate} \quad \text{(parent)} \quad \text{apple} \quad \text{(child)} \]

\[ \text{ate} \rightarrow [0.6, 0.9, 0.3, -1.0, 0.1, -0.7] \]

\[ \text{apple} \rightarrow [-0.9, 0.1, -0.3, 0.5, 0.1, 0.6] \]
Motivation

For all of the experiments in this paper, we used the Liang (2005) implementation of the Brown algorithm to obtain the necessary word clusters.

3 Feature design

Key to the success of our approach is the use of features which allow word-cluster-based information to assist the parser. The feature sets we used are similar to other feature sets in the literature (McDonald et al., 2005a; Carreras, 2007), so we will not attempt to give an exhaustive description of the features in this section. Rather, we describe our features at a high level and concentrate on our methodology and motivations. In our experiments, we employed two different feature sets: a baseline feature set which draws upon “normal” information sources such as word forms and parts of speech, and a cluster-based feature set that also uses information derived from the Brown cluster hierarchy.

3.1 Baseline features

Our first-order baseline feature set is similar to the feature set of McDonald et al. (2005a), and consists of indicator functions for combinations of words and parts of speech for the head and modifier of each dependency, as well as certain contextual tokens.

Our second-order baseline features are the same as those of Carreras (2007) and include indicators for triples of part of speech tags for sibling interactions and grandparent interactions, as well as additional bigram features based on pairs of words involved in these higher-order interactions. Examples of baseline features are provided in Table 1.

We augment the McDonald et al. (2005a) feature set with backed-off versions of the “Surrounding Word POS Features” that include only one neighboring POS tag. We also add binned distance features which indicate whether the number of tokens between the head and modifier of a dependency is greater than 2, 5, 10, 20, 30, or 40 tokens.

3.2 Cluster-based features

The first- and second-order cluster-based feature sets are supersets of the baseline feature sets: they include all of the baseline feature templates, and add an additional layer of features that incorporate word clusters. Following Miller et al. (2004), we use prefixes of the Brown cluster hierarchy to produce clusterings of varying granularity. We found that it was nontrivial to select the proper prefix lengths for the dependency parsing task; in particular, the prefix lengths used in the Miller et al. (2004) work (between 12 and 20 bits) performed poorly in dependency parsing.

After experimenting with many different feature configurations, we eventually settled on a simple but effective methodology. First, we found that it was helpful to employ two different types of word clusters:

1. Short bit-string prefixes (e.g., 4–6 bits), which we used as replacements for parts of speech.

One possible explanation is that the kinds of distinctions required in a named-entity recognition task (e.g., “Alice” versus “Intel”) are much finer-grained than the kinds of distinctions relevant to syntax (e.g., “apple” versus “eat”).

The table below shows examples of baseline and cluster-based feature templates. Each entry represents a class of indicators for tuples of information. For example, “ht,mt” represents a class of indicator features with one feature for each possible combination of head POS-tag and modifier POS-tag. Abbreviations:

ht = head POS, hw = head word, hc4 = 4-bit prefix of head, hc6 = 6-bit prefix of head, hc* = full bit string of head; mt, mw, mc4, mc6, mc* = likewise for modifier; st, gt, sc4, gc4, ... = likewise for sibling and grandchild.

(McDonald et al., 2005; Koo et al., 2008)
Motivation

- Train link embeddings on tons of auto-parsed data
  *(min-count thresholded to get only the popular links)*

\[\begin{align*}
ate & \quad (\text{parent}) \\
\rightarrow & \\
apple & \quad (\text{child})
\end{align*}\]
Motivation

( unary )

\[ w[i] \rightarrow 0.4 \]

\[ ate \quad (\text{parent}) \]

\[ apple \quad (\text{child}) \]

\[ ate - apple \rightarrow [0.4, 0.2, -0.3, 0.4, 0.1, 0.7] \]
Motivation

(arity)

prefix6 \to 101010
prefix4 \to 1101

\( \text{ate} \)
\( \text{(parent)} \)
\( \text{apple} \)
\( \text{(child)} \)

\( \text{ate} - \text{apple} \to 11011100010 \)

\( \text{prefix4} \)
\( \text{prefix6} \)

00
ate—apple

01
ate—pear

10
Cole—Mr.

11
Cruz—Mr.
Motivation

boost—revenue

tap—market

reduce—holdings

attract—investment

build—business

boosting—bid

allocating—$

taking—shares

having—losses

N.Y.—Yonkers

N.Y.—Bronx

Md.—Columbia

Ky.—Lexington

Va.—Reston

only—1.5

about—170

almost—15

roughly—50

Swete—Mr.

Case—Mr.

Mantoya—Mr.

Leny—Mr.
Motivation

dense syntactic features

\[
\begin{align*}
\text{ate—apple:} & & [0.2, -0.9, 0.4, -0.1, 0.3] \\
\text{ate—pear:} & & [0.6, -0.2, 0.1, -0.4, 0.1] \\
\text{had—water:} & & [0.1, 0.9, -0.2, -0.1, 0.5] \\
& & \ldots \\
& & \ldots \\
& & \ldots \\
\text{from—home:} & & [0.1, -0.9, 0.2, -0.8, 0.7] \\
\text{under—tree:} & & [0.9, -0.4, 0.4, 0.5, -0.3] \\
\text{for—her:} & & [-0.3, 0.9, -0.4, 0.2, 0.2]
\end{align*}
\]
Motivation

- Much fewer, simpler **unary** features vs. millions of template-based, **n-ary** word-cluster features.

- Directly work with higher-order, substructure embeddings that task factors on, and their hidden relationships.

- Portable as off-the-shelf, dense, syntactic features (instead of lexicalized or word embedding features).
Training

- Parse a large corpus with baseline parser
- Tuples consist of a dependency link and its context
  (Bansal et al., 2014; Levy and Goldberg, 2014)

... said that the regulation of the internet is ...
Training

- Tuples consist of a dependency link and its context

\[
\begin{array}{cccccc}
\text{dist.} & \text{gp label} & \text{parent–child link} & \text{dep label} & \text{dist.} \\
-2 & NMOD_{GL} & \text{of—internet} & PMOD_{L} & -2
\end{array}
\]
Training

- Tuples consist of a dependency link and its context

```
.gl<GL> gp—p p—c d<D> l<L>
```

gp label   gp-parent link   parent-child link   dist.   dep label

```
[NMOD<GL> regulation—of of—internet -2 PMOD<L>]
```

collection window

- Run SKIP-gram model to predict context
- Threshold of 4 to get a vocab of just 92K
  (and then backoff to word/unk features)
Training

Tried other context versions, e.g., for link to be able to predict the individual words (for backing off to word-level shared info)

\[
[NMOD_{<GL>} \text{ regulation--of of--internet internet -2 PMOD}_{<L>}] 
\]
Clusters

Finds useful groups and subtle distinctions at link level


[began–Meanwhile, was–Since, are–Often, would–Now, had–During, were–Over, was–Late, have–Until, ...]
Clusters

- Finds useful groups and subtle distinctions at link level

\[\text{says–mean, adds–may, explains–have, contend–has, recalls–had, figures–is, asserted–is, notes–would, ...}\]

\[\text{would–Based, is–Besides, was–Like, is–From, are–Despite, said–Besides, says–Despite, reported–As, ...}\]

\[\text{Catsimatidis–Mr., Swete–Mr., Case–Mr., Montoya–Mr., Byerlein–Mr., Heard–Mr., Leny–Mr., Graham–Mrs., ...}\]

\[\text{only–1.5, about–170, nearly–eight, approximately–10, almost–15, some–80, Only–two, about–23, roughly–50, ...}\]
Dependency Parsing Features

Brown cluster n-ary features (Koo et al., 2008):

prefix6 \rightarrow 110010
prefix4 \rightarrow 1100

\textit{tag} \rightarrow \textit{VBD} \rightarrow \textit{NN}

\text{apple} \rightarrow 00010100010

\textbf{prefix4}

\textbf{prefix6}

\text{apple}

\text{(parent)}

\text{ate}

\text{(child)}

(McDonald et al., 2005; Koo et al., 2008)
Dependency Parsing Features

- **Word embedding n-ary features** (Bansal et al., 2014):
  - Per-dimension bucket features:

  \[
  \text{ate} \rightarrow [0.2 \ 0.7 \ -0.6 \ 0.9] \\
  \text{apple} \rightarrow [0.6 \ -0.1 \ 0.7 \ 0.2]
  \]

- **Hierarchical clustering (bit string) features:**

  \[\text{linkage}(E, \text{‘ward’, ‘euclidean’})\]

  \[
  \text{apple} \rightarrow 00010100010
  \]

  (McDonald et al., 2005; Koo et al., 2008)
Dependency Parsing Features

- **Link embedding **unary **features** (this work):
  - Per-dimension bucket features:
    - \( ate \rightarrow \text{apple} \rightarrow [-0.5 \ 0.3 \ 0.6 \ 0.8] \)
    - \( ate \rightarrow \text{apple} \) w/ \( \text{dim}=3 \):

- **Hierarchical clustering (bit string) features:**
  - \( \text{linkage}(E, \text{‘ward’}, \text{‘euclidean’}) \)
  - \( ate \rightarrow \text{apple} \rightarrow 11011100010 \)
Feature Comparison (Memory, Speed)

- Setup: MSTParser (2\textsuperscript{nd} order), standard data splits, parameters, preprocessing, threshold (Bansal et al., 2014)
- Much fewer features compared to n-ary, word-based
- Quicker to train these SKIP-based link features

<table>
<thead>
<tr>
<th>System</th>
<th>Number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>5M</td>
</tr>
<tr>
<td>BROWN</td>
<td>13M</td>
</tr>
<tr>
<td>Bansal et al. (2014)</td>
<td>30M</td>
</tr>
<tr>
<td>Bucket Bit-string</td>
<td>15K</td>
</tr>
<tr>
<td></td>
<td>1M</td>
</tr>
</tbody>
</table>

(2.5 days)
(15 mins.)
(1 day)
Dependency Parsing Results

- Stat-equal improvements as Brown and stat-significant stacking, at much fewer, simpler, quicker features

<table>
<thead>
<tr>
<th>System</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>91.9</td>
</tr>
<tr>
<td>+ BROWN</td>
<td>92.7</td>
</tr>
<tr>
<td>+ Bucket</td>
<td>92.3</td>
</tr>
<tr>
<td>+ Bit-string</td>
<td>92.6</td>
</tr>
<tr>
<td>+ BROWN + Bucket</td>
<td>93.0</td>
</tr>
<tr>
<td>+ BROWN + Bit-string</td>
<td>93.1</td>
</tr>
</tbody>
</table>

- Similar improvements and stacking for out-of-domain
- Allows practical, accurate per-dimension features
Off-the-shelf Results

- Portable as simple, dense, syntactic features
- E.g., on constituent reranking, stat-equal improvements as global reranking features and stat-signif. stacking (Bansal and Klein, 2011)

<table>
<thead>
<tr>
<th>Parsing Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
</tr>
<tr>
<td>Baseline (1-best)</td>
<td>90.2</td>
</tr>
<tr>
<td>Baseline (log ( p(t</td>
<td>w) ))</td>
</tr>
<tr>
<td>+ Config</td>
<td>91.1</td>
</tr>
<tr>
<td>+ Bit-string</td>
<td>90.9</td>
</tr>
<tr>
<td>+ Config + Bit-string</td>
<td>91.4</td>
</tr>
</tbody>
</table>
Conclusion

- Dense dependency link embeddings allow simpler, fewer (unary) features in dependency parsing

- Get similar improvements to n-ary template-based word cluster/embedding features

- Portable as useful, dense, syntactic features to downstream tasks, e.g., constituent reranking
Current/Future Work

- Export to other extrinsic tasks, e.g., sentence classification or initial units in vector-space composition
- Newer, better representation learning tools, e.g., deeper NNs, GloVe (Pennington et al., 2014)
- Training on larger quantities of automatically-parsed data
- Other back-off approaches for unknown links, e.g., tag-based links (tag1—tag2) and one-sided links (UNK-word or UNK-tag)
- Compare to approaches like relation matrices on dependency labels, concatenation+SVD of head and argument vectors
Thank you!

Data (link embeddings and features) at:

ttic.uchicago.edu/~mbansal/codedata/linkEmbeddings-skiplink.zip