FINDING FUNCTION IN FORM

Chris Dyer

Joint work with:
Miguel Ballesteros
Kazuya Kawakami
Wang Ling
Noah A. Smith
Linguistic representation

**ARBITRARINESS** (de Saussure, 1916)

\[ \text{car} - \text{c} + \text{b} = \text{bar} \]

\[ \text{cat} - \text{c} + \text{b} = \text{bat} \]

**COMPOSITIONALITY** (Frege, 1892)

John dances - John + Mary = Mary dances

\[ \text{DANCE} (\text{JOHN}) \]

John sings - John + Mary = Mary sings

\[ \text{SING} (\text{JOHN}) \]
Linguistic representation

CHALLENGE 1: IDIOMS

John saw the football
John saw the bucket
John kicked the football
John kicked the bucket

CHALLENGE 2: MORPHOLOGY

cool | coooool | coooooooooool

cat + s = cats
bat + s = bats
Linguistic representation

- **Arbitrariness** and **compositionality** exist at all levels—although the tendency is smaller units are more arbitrary and larger units are more compositional.

- Learners must be able to do both

- Strategy in this work: use LSTMs (nominally a model with good generalization performance) at the lexical level where we usually use “memorization”
Morphological typology
A crash course

Languages have different amounts of morphology:

**Analytic**
- Mandarin
- English

**Fusional**
- Spanish
- Russian

**Agglutinative**
- Turkish
- Hungarian

**Polysynthetic**
- Inuktitut

**Templatic**
- Arabic
- Hebrew
- Korean(?)
- Finnish
- Swahili
- Mohawk
- Chukchi

Increasing Lexical Complexity
Morphological typology
A crash course

Languages have different amounts of morphology:

- **Analytic**
  - “cat”

- **Fusional**
  - “to the cat”

- **Agglutinative**
  - “they caused it not to meow”

- **Polysynthetic**
  - “they would carry the cat on their shoulders”
Talk outline

• RNNs and Two word embedding models

• Two applications (“main course”)  
  • Dependency parsing with Stack LSTMs  
  • Language Modeling

• Some preliminary work (“desert”)  
  • Text-Color Regression  
  • Translation

• Summary
What is a vector representation of a sequence $\mathcal{x}$?

Note: numerous definitions exist for $f$. 

Recurrent Neural Networks (RNNs)

$$c = \text{RNN}(\mathcal{x})$$

$$\mathcal{x} = \text{START} \ x_1 \ x_2 \ x_3 \ x_4$$

$$h_t = f(h_{t-1}, x_t)$$
while $y_t \neq \text{STOP}$
\[
\begin{align*}
    h_t &= f(h_{t-1}, x_t) \\
    y_t &\sim g(h_t) \\
    t &\leftarrow t + 1
\end{align*}
\]

What is the probability of a sequence $y$?
Word Embedding Models

Memorize

START  c  a  r  STOP

Generalize
Talk outline

• RNNs and Two word embedding models

• Two applications (“main course”)
  • Dependency parsing with Stack LSTMs
  • Language Modeling

• Some preliminary work (“desert”)
  • Text-Color Regression
  • Translation

• Summary
Example
Dependency parsing

I saw her duck

ROOT

I saw her duck
Transition-based parsing

- Build trees by pushing words ("shift") onto a stack and combing elements at the top of the stack into a syntactic constituent ("reduce")

- Given current stack and buffer of unprocessed words, what action should the algorithm take?

- Widely used
  - Good accuracy
  - $O(n)$ runtime [much faster than other parsing algos]
<table>
<thead>
<tr>
<th>Stack</th>
<th>Buffer</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I saw her duck</td>
<td>SHIFT</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>SHIFT</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>REDUCE-L</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>SHIFT</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>REDUCE-L</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>REDUCE-R</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>SHIFT</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td>REDUCE-R</td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I saw her duck</td>
<td></td>
</tr>
</tbody>
</table>
Transition-based parsing

Challenges

unbounded depth

unbounded length

arbitrarily complex trees

reading and forgetting

unbounded history
Transition-based parsing

Solutions

- Use a new variant of LSTMs—stack LSTMs—to embed buffer, stack, and history of actions

  - Embeddings are sensitive to full lookahead, full stack contents, and full history of actions

  - Incremental construction of parser state embeddings means runtime remains linear
Transition-based parsing

Stack LSTMs

• Augment LSTM with a stack pointer

• Two constant-time operations
  • **Push** - read input, add to top of stack
  • **Pop** - move stack pointer back

• A **summary** of stack contents is obtained by accessing the output of the LSTM at location of the stack pointer
Transition-based parsing

Stack LSTMs

\[ y_0 \]

\[ \emptyset \]

PUSH
Transition-based parsing
Stack LSTMs

POP
Transition-based parsing
Stack LSTMs

\[ y_0 \rightarrow \emptyset \rightarrow x_1 \rightarrow y_1 \]
PUSH
Transition-based parsing
Stack LSTMs

POP
Transition-based parsing

Stack LSTMs

\( y_0 \)
\( \emptyset \)
\( y_1 \)
\( x_1 \)
\( y_2 \)
\( x_2 \)

PUSH
Transition-based parsing
Stack LSTMs
\[ p_t \]
an overhasty decision
overhasty decision was made
An overhasty decision was made.
Transition-based parsing

Experimental setup

Word Lookup

CharLSTM
Transition-based parsing
CharLSTM > Word Lookup

<table>
<thead>
<tr>
<th>Language</th>
<th>Word</th>
<th>Chars</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>91.2</td>
<td>91.5</td>
<td>+0.3</td>
</tr>
<tr>
<td>Turkish</td>
<td>71.7</td>
<td>76.3</td>
<td>+4.6</td>
</tr>
<tr>
<td>Hungarian</td>
<td>72.8</td>
<td>80.4</td>
<td>+7.6</td>
</tr>
<tr>
<td>Basque</td>
<td>77.1</td>
<td>85.2</td>
<td>+8.1</td>
</tr>
<tr>
<td>Korean</td>
<td>78.7</td>
<td>88.4</td>
<td>+9.7</td>
</tr>
<tr>
<td>Swedish</td>
<td>76.4</td>
<td>79.2</td>
<td>+3.2</td>
</tr>
<tr>
<td>Swedish</td>
<td>76.4</td>
<td>79.2</td>
<td>+2.8</td>
</tr>
<tr>
<td>Arabic</td>
<td>85.2</td>
<td>86.1</td>
<td>+0.9</td>
</tr>
<tr>
<td>Chinese</td>
<td>79.1</td>
<td>79.9</td>
<td>+0.8</td>
</tr>
</tbody>
</table>

In English parsing, the character LSTM is roughly equivalent to the lookup approach.

What about languages with richer lexicons?

Turkish: Muvaffakiyetsizleştiricileştirirveremeyebileceklerimizdenmişsinizcesine
Hungarian: Megszentségteleníthetetlenségeskedéseitekért
Transition-based parsing
CharLSTM > Word Lookup

<table>
<thead>
<tr>
<th>Language</th>
<th>Word</th>
<th>Chars</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>91.2</td>
<td>91.5</td>
<td>+0.3</td>
</tr>
</tbody>
</table>
Transition-based parsing
CharLSTM > Word Lookup

<table>
<thead>
<tr>
<th>Language</th>
<th>Word</th>
<th>Chars</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>91.2</td>
<td>91.5</td>
<td>+0.3</td>
</tr>
<tr>
<td>Turkish</td>
<td>71.7</td>
<td>76.3</td>
<td>+4.6</td>
</tr>
<tr>
<td>Hungarian</td>
<td>72.8</td>
<td>80.4</td>
<td>+7.6</td>
</tr>
<tr>
<td>Basque</td>
<td>77.1</td>
<td>85.2</td>
<td>+8.1</td>
</tr>
<tr>
<td>Korean</td>
<td>78.7</td>
<td>88.4</td>
<td>+9.7</td>
</tr>
</tbody>
</table>

In **agglutinative** languages,
Transition-based parsing
CharLSTM > Word Lookup

<table>
<thead>
<tr>
<th>Language</th>
<th>Word</th>
<th>Chars</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>91.2</td>
<td>91.5</td>
<td>+0.3</td>
</tr>
<tr>
<td>Turkish</td>
<td>71.7</td>
<td>76.3</td>
<td>+4.6</td>
</tr>
<tr>
<td>Hungarian</td>
<td>72.8</td>
<td>80.4</td>
<td>+7.6</td>
</tr>
<tr>
<td>Basque</td>
<td>77.1</td>
<td>85.2</td>
<td>+8.1</td>
</tr>
<tr>
<td>Korean</td>
<td>78.7</td>
<td>88.4</td>
<td>+9.7</td>
</tr>
<tr>
<td>Swedish</td>
<td>76.4</td>
<td>79.2</td>
<td>+3.2</td>
</tr>
<tr>
<td>Swedish</td>
<td>76.4</td>
<td>79.2</td>
<td>+2.8</td>
</tr>
<tr>
<td>Arabic</td>
<td>85.2</td>
<td>86.1</td>
<td>+0.9</td>
</tr>
</tbody>
</table>

In **agglutinative** languages,

In **fusional/templatic** languages,
## Transition-based parsing

**CharLSTM > Word Lookup**

<table>
<thead>
<tr>
<th>Language</th>
<th>Word</th>
<th>Chars</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>91.2</td>
<td>91.5</td>
<td>+0.3</td>
</tr>
<tr>
<td>Turkish</td>
<td>71.7</td>
<td>76.3</td>
<td>+4.6</td>
</tr>
<tr>
<td>Hungarian</td>
<td>72.8</td>
<td>80.4</td>
<td>+7.6</td>
</tr>
<tr>
<td>Basque</td>
<td>77.1</td>
<td>85.2</td>
<td>+8.1</td>
</tr>
<tr>
<td>Korean</td>
<td>78.7</td>
<td>88.4</td>
<td>+9.7</td>
</tr>
<tr>
<td>Swedish</td>
<td>76.4</td>
<td>79.2</td>
<td>+3.2</td>
</tr>
<tr>
<td>Swedish</td>
<td>76.4</td>
<td>79.2</td>
<td>+2.8</td>
</tr>
<tr>
<td>Arabic</td>
<td>85.2</td>
<td>86.1</td>
<td>+0.9</td>
</tr>
<tr>
<td>Chinese</td>
<td>79.1</td>
<td>79.9</td>
<td>+0.8</td>
</tr>
</tbody>
</table>

In **agglutinative** languages,

In **fusional/templatic** languages,

In **analytic** languages, the models are roughly equivalent.
Transition-based parsing

OOV rate vs. error reduction
Transition-based parsing

What is really being learned?

• Is it fair to say we are learning anything **beyond syntactic classes** (which are well-known to have morphological reflexes)?

• Let’s turn to an even more basic task—**language modeling**
Language modeling

LSTM language model
Language modeling
Experimental setup

What input representation?

Evaluate on held-out perplexity (and # of parameters).
Language modeling
CharLSTM > Word Lookup

<table>
<thead>
<tr>
<th>Language</th>
<th>ppl Words</th>
<th>ppl Chars</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>59.4</td>
<td>57.4</td>
<td>-2.0</td>
</tr>
</tbody>
</table>

Analytic

Agglutinative

Fusional
## Language modeling

### Word similarities

<table>
<thead>
<tr>
<th>increased</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduced</td>
<td>Richard</td>
</tr>
<tr>
<td>improved</td>
<td>George</td>
</tr>
<tr>
<td>expected</td>
<td>James</td>
</tr>
<tr>
<td>decreased</td>
<td>Robert</td>
</tr>
<tr>
<td>targeted</td>
<td>Edward</td>
</tr>
</tbody>
</table>
Character vs. word modeling

Summary

- Model performance is essentially equivalent in morphologically simple languages (e.g., Chinese, English)

- In morphologically rich languages (e.g., Hungarian, Turkish, Finnish), performance improvements are most pronounced

- We need *far fewer parameters* to represent words as “compositions” of characters
Talk outline

• RNNs and Two word embedding models

• Two applications ("main course")
  • Dependency parsing with Stack LSTMs
  • Language Modeling

• Some preliminary work ("desert")
  • Text-Color Regression
  • Translation

• Summary
Word–color Regression

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>blue</td>
<td>blue</td>
</tr>
<tr>
<td>red</td>
<td>red</td>
</tr>
<tr>
<td>green</td>
<td>green</td>
</tr>
<tr>
<td>black</td>
<td>black</td>
</tr>
<tr>
<td>purple</td>
<td>purple</td>
</tr>
</tbody>
</table>
## Word–color Regression

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>blue</td>
<td>blue</td>
<td>carrot</td>
<td>carrot</td>
</tr>
<tr>
<td>red</td>
<td>red</td>
<td>firw</td>
<td>firw</td>
</tr>
<tr>
<td>green</td>
<td>green</td>
<td>Artemis</td>
<td>Dark</td>
</tr>
<tr>
<td>black</td>
<td>black</td>
<td>Divinely Pink</td>
<td>Divinely Pink</td>
</tr>
<tr>
<td>purple</td>
<td>purple</td>
<td>emo love</td>
<td>emo love</td>
</tr>
</tbody>
</table>

**Figure**: A table illustrating the mapping of words to colors. Each row shows a word $x$ paired with a corresponding color $y$. The right side of the figure shows the same mapping for another set of words and colors.
Word–color Regression

Q: How to represent colors and measure color distances?

RGB colorspace: convenient, but perceptually nonuniform

Use Lab (CIELUV) instead

Perceptual nonuniformity in CXY colorspace
## Word–color Regression: Held-out predictions

<table>
<thead>
<tr>
<th>Reference</th>
<th>CharLSTM</th>
<th>Reference</th>
<th>CharLSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>bacon lipstick</td>
<td>bacon lipstick</td>
<td>Speedcap</td>
<td>Speedcap</td>
</tr>
<tr>
<td>bensada</td>
<td>bensada</td>
<td>Kaylee</td>
<td>Kaylee</td>
</tr>
<tr>
<td>Violet shadow</td>
<td>Violet shadow</td>
<td>mint pint</td>
<td>mint pint</td>
</tr>
<tr>
<td>pineapple twist</td>
<td>pineapple twist</td>
<td>flushed lips</td>
<td>flushed lips</td>
</tr>
<tr>
<td>night drama</td>
<td>night drama</td>
<td>prior</td>
<td>prior</td>
</tr>
<tr>
<td>SunnyGlow</td>
<td>SunnyGlow</td>
<td>Rose Violet</td>
<td>Rose Violet</td>
</tr>
<tr>
<td>Pink Scarf</td>
<td>Pink Scarf</td>
<td>Child Cake</td>
<td>Child Cake</td>
</tr>
<tr>
<td>wet stadium grass</td>
<td>wet stadium grass</td>
<td>in December</td>
<td>in December</td>
</tr>
</tbody>
</table>
Word–color Regression: Analysis

- Color “semantics” is simple, and perceptual facts are well understood (also: lots of data!)

- Color names in our dataset are mostly arbitrary—but somewhat compositional in places

- Summary: we are able to learn to map from color words to “perceptual semantics” with LSTMs
Preliminary work
Translation modeling
Preliminary work
Translation modeling
Preliminary work

Translation modeling

Beginnings

START

Aller  Anfang  ist  schwer  STOP
Preliminary work

Translation modeling

Beginnings are

START

Aller Anfang ist schwer STOP
Preliminary work

Translation modeling

Beginnings are difficult

START

Aller Anfang ist schwer STOP
Preliminary work
Translation modeling

Beginnings are difficult STOP

START

Aller Anfang ist schwer STOP

Preliminary work
Translation modeling

Beginnings are difficult STOP

START

Aller Anfang ist schwer STOP
Preliminary work
Translation modeling

Beginnings
Preliminary work
Translation modeling
Preliminary work
Translation modeling

Europe is not just a sum of Member States.

A Europa não é a uma sustinada dos Estados-Membros.

“sustinated”?

Mr President, I voted for the Murphy report.

Senhor Presidente, votei a favor do relatório Soder.
Final Thoughts

• (Bidirectional) LSTMs are an effective means to model both the regularity and arbitrariness of the lexicon

• This raises the questions: do we need words at all? Should the existence of words be learned?
thanks!

questions?
Representing Tree(let)s

an overhasty decision
Representing Tree(let)s

an overhasty decision

an overhasty decision