Applications of Deep Learning for NLP

Dr. Kira Radinsky
CTO SalesPredict
Visiting Professor/Scientist Technion

Slides were adapted from lecture by Richard Socher
Overview

- Model overview: How to properly compare to other models and choose your own model
  - Word representation
  - Phrase composition
  - Objective function
  - Optimization

- Character RNNs on text and code
- Morphology
- Logic
- Question Answering
- Image – Sentence mapping
Model overview: Word Vectors

• Random

• Word2Vec

• Glove

• Dimension – often defines the number of model parameters

• Or work directly on characters or morphemes
Model overview: Phrase Vector Composition

- Composition Function governs how exactly word and phrase vectors interact to compose meaning

- Averaging: \( p = a + b \)
  - Lots of simple alternatives

- Recursive neural networks

- Convolutional neural networks

- Recurrent neural network
Composition: Bigram and Recursive functions

- Many related models are special cases of MV–RNN
  \[ p = f \left( W \begin{bmatrix} Ba \\ Ab \end{bmatrix} \right) \]

- Mitchell and Lapata, 2010; Zanzotto et al., 2010:
  \[ p = Ba + Ab = id \left( \begin{bmatrix} I_{n \times n} \end{bmatrix} \begin{bmatrix} Ba \\ Ab \end{bmatrix} \right) \]

- Baroni and Zamparelli (2010): A is an adjective matrix and b is a noun vector
  \[ p = Ab = id \left( \begin{bmatrix} 0_{n \times n} \\ I_{n \times n} \end{bmatrix} \begin{bmatrix} Ba \\ Ab \end{bmatrix} \right) \]

- RNNs of Socher et al. 2011 (ICML, EMNLP, NIPS) are also special cases
  \[ p = f \left( W \begin{bmatrix} I_{n \times n}a \\ I_{n \times n}b \end{bmatrix} \right) \]

- **Recursive neural tensor** networks bring quadratic and multiplicative interactions between vectors
Additional choice for recursive neural nets

- Dependency trees focus more on semantic structure

1. Constituency Tree

2. Dependency Tree

3. Balanced Tree
Discussion: Compositional Structure

- Recursive Neural Networks so far used constituency trees which results in more syntactically influenced representations.

- Instead: Use dependency trees which capture more semantic structure.
Discussion: Compositional Structure

A constituency parse tree breaks a text into sub-phrases. Non-terminals in the tree are types of phrases, the terminals are the words in the sentence, and the edges are unlabeled.

A dependency parse connects words according to their relationships. Each vertex in the tree represents a word, child nodes are words that are dependent on the parent, and edges are labeled by the relationship.
Composition: CNNs

- Several variants also:
  - No pooling layers
  - Pooling layers: simple max--pooling or dynamic pooling
  - Pooling across different dimensions

- Somewhat less explored in NLP than RNNs$^2$

- Not linguistically nor cognitively plausible
Composition: Recurrent Neural Nets

- Vanilla
- GRU
- LSTM
- Many variants of LSTMs
  “LSTM: A Search Space Odyssey” by Greff et al. 2015
Model overview: Objective function

- **Max-margin**

- **Cross-entropy**
  - Supervised to predict a class
  - Unsupervised: predict surrounding words

- **Auto-encoder**
  - My opinion: Unclear benefits for NLP
  - Unless encoding another modality
Auto Encoders

How well can we re-create from the new representation the original one?
Optimization

• Initialization (word vector and composition parameters)!!

• Optimization algorithm
  • SGD
  • SGD + momentum (a good default to start with!)
  • L-BFGS
  • AdaGrad (a good default if you have word vectors that you are learning)
  • Adelta

• Optimization tricks
  • Regularization (some define as part of model)
  • Dropout
Character RNNs on text and code

http://karpathy.github.io/2015/05/21/rnn---effectiveness/
Character RNNs on text and code

- Haven’t yet produced useful results on real datasets

- Shows that RNNs can memorize sequences and keep memory (mostly LSTMs)

- Most interesting results simply train on dataset and sample from it afterwards (first shown by Sutskever et al. 2011: Generating Text with Recurrent Neural Networks)

- Results from an LSTM (karpathy.github.io) →
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.
Naturalism and decision for the majority of Arab countries' capital ide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict.
Proof. Omitted. □

Lemma 0.1. Let $C$ be a set of the construction.

Let $C$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{O_X} = \mathcal{O}_X(\mathcal{L})$$

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\text{etale}}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} \mathcal{G}, \mathcal{F}\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. This is an integer $\mathbb{Z}$ is injective.

Proof. See Spaces, Lemma ???. □

Lemma 0.3. Let $S$ be a scheme. Let $X$ be a scheme and $X$ is an affine open covering. Let $U \subset X$ be a canonical and locally of finite type. Let $X$ be a scheme. Let $X$ be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let $X$ be a scheme. Let $X$ be a scheme covering. Let

$$b : X \to Y' \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over $S$ and $Y$.

Proof. Let $X$ be a nonzero scheme of $X$. Let $X$ be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over $S$.

(2) If $X$ is an affine open covering.

Consider a common structure on $X$ and $X$ the functor $\mathcal{O}_X(U)$ which is locally of finite type.

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{array}{ccc}
S & \to & X' \\
\downarrow & & \downarrow \\
\mathcal{G} & \to & \mathcal{F}
\end{array}$$

is a limit. Then $\mathcal{G}$ is a finite type and assume $S$ is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type $f_*$. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

Proof. We have see that $X = \text{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of $X$ is an open neighbourhood of $U$. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ???. A reduced above we conclude that $U$ is an open covering of $C$. The functor $\mathcal{F}$ is a “field”

$$\mathcal{O}_{X,x} \to \mathcal{F}_x \to \mathcal{O}_{X,x} \times \mathcal{O}_{X,x}$$

is an isomorphism of covering of $\mathcal{O}_{X,x}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that $X$ is an isomorphism.

The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over $S$.

If $\mathcal{F}$ is a scheme theoretic image points.

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X,x}$ is a closed immersion, see Lemma ???. This is a sequence of $\mathcal{F}$ is a similar morphism.
Code! (Linux source code)

/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */

static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
So why are not we using this?

1. Does not necessarily give us the best semantic word vector representations that can be useful for downstream tasks
2. We are quite far in research from getting to semantically meaningful representation and not just syntactic
Morphology

• Better Word Representations with Recursive Neural Networks for Morphology – Luong et al. (slides from Luong)

• *Problem with word vectors*: poorly estimate rare and complex words.

<table>
<thead>
<tr>
<th></th>
<th>(Collobert &amp; Weston, 2010)</th>
<th>(Huang et. al., 2012)</th>
</tr>
</thead>
<tbody>
<tr>
<td>distinct</td>
<td>different distinctive broader narrower</td>
<td>unique broad distinctive separate</td>
</tr>
<tr>
<td>distinctness</td>
<td>morphologies pesawat clefts pathologies</td>
<td>companion roskam hitoshi enjoyed</td>
</tr>
<tr>
<td>affect</td>
<td>exacerbate impacts characterize</td>
<td>allow prevent involve enable</td>
</tr>
<tr>
<td>unaffected</td>
<td>unnoticed dwarfed mitigated</td>
<td>monti sheaths krystal</td>
</tr>
</tbody>
</table>
Limitations of existing work

• Treat related words as independent entities.
  • There is no real connection between distinct, distinctive, distinctively, etc.
• Represent unknown words with a few vectors.

Word frequencies in Wikipedia docs (986m tokens)
Luong’s approach – Context-sensitive morphological RNN

- **Neural Language Model**: simple feed-forward network (Huang, et al., 2012) with ranking-type cost (Collobert et al., 2011).
- **Morphology Model**: recursive neural network (Socher et al., 2011).

\[ v^T f(W[x_1; x_2; \ldots; x_n] + b) \]

\[ p = f(W_m[x_{stem}; x_{affix}] + b_m) \]
**Morphological Recursive NN**

- Assumes access to a dictionary of morphemic analyses of words
  - Use Morfessor
  - Distinct morphemes are encoded by column vectors

![Diagram](image)
Analysis

• Blends word structure and syntactic-semantic information.

<table>
<thead>
<tr>
<th>Words</th>
<th>(Collobert et al., 2011)</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>commenting</td>
<td>insisting insisted focusing</td>
<td>commented comments criticizing</td>
</tr>
<tr>
<td>unaffected</td>
<td>unnoticed dwarfed mitigated</td>
<td>undesired unhindered unrestricted</td>
</tr>
<tr>
<td>distinct</td>
<td>different distinctive broader</td>
<td>divergent diverse distinctive</td>
</tr>
<tr>
<td>distinctness</td>
<td>morphologies pesawat clefts</td>
<td>distinctiveness smallness largeness</td>
</tr>
<tr>
<td>heartlessness</td>
<td>$\emptyset$</td>
<td>corruptive inhumanity ineffectual</td>
</tr>
<tr>
<td>Saudi-owned</td>
<td>avatar mohajir kripalani</td>
<td>saudi-based syrian-controlled</td>
</tr>
</tbody>
</table>
Solutions to the problem of polysemous words

- Improving Word Representations Via Global Context And Multiple Word Prototypes by Huang et al. 2012
Natural language inference

Claim: Simple task to define, but engages the full complexity of compositional semantics:

- Lexical entailment
- Quantification
- Coreference
- Lexical/scope ambiguity
- Commonsense knowledge
- Propositional attitudes
- Modality
- Factivity and implicativity
First training data

• **Training data:**
  - dance *entails* move
  - waltz *neutral* tango
  - tango *entails* dance
  - sleep *contradicts* dance
  - waltz *entails* dance

Memorization (training set): Generalization (test set):

• dance ??? move 
  - sleep ??? waltz
• waltz ??? tango 
  - tango ??? move
### Natural language inference: definitions!

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \equiv y$</td>
<td>equivalence</td>
<td>$\text{couch} \equiv \text{sofa}$</td>
</tr>
<tr>
<td>$x \sqsubset y$</td>
<td>forward entailment (strict)</td>
<td>$\text{crow} \sqsubset \text{bird}$</td>
</tr>
<tr>
<td>$x \sqsupset y$</td>
<td>reverse entailment (strict)</td>
<td>$\text{European} \sqsupset \text{French}$</td>
</tr>
<tr>
<td>$x^{\perp} y$</td>
<td>negation (exhaustive exclusion)</td>
<td>$\text{human}^{\perp} \text{nonhuman}$</td>
</tr>
<tr>
<td>$x \mid y$</td>
<td>alternation (non-exhaustive exclusion)</td>
<td>$\text{cat} \mid \text{dog}$</td>
</tr>
<tr>
<td>$x \sqcap y$</td>
<td>cover (exhaustive non-exclusion)</td>
<td>$\text{animal} \sqcap \text{nonhuman}$</td>
</tr>
<tr>
<td>$x # y$</td>
<td>independence</td>
<td>$\text{hungry} # \text{hippo}$</td>
</tr>
</tbody>
</table>

---

*Slide from Bill MacCartney*
A minimal NN for lexical relations

- Words are learned embedding vectors.
- One plain RNN or RNTN layer
- Softmax emits relation labels
- Learn everything with SGD.
Recursion in propositional logic

Experimental approach: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:
• Learn the relations between individual words. (lexical relations)
• Learn how lexical relations impact phrasal relations. (composition)
  • This needs to be recursively applicable!

• $a \equiv a$, $a \land (\neg a)$, $a \equiv (\neg \neg a)$, ...
Natural language inference with RNNs

- Two trees + learned comparison layer, then a classifier:
Natural language inference with RNNs

Accuracy vs. Size of longer expression
- 25d TreeRNTN
- 45d TreeRNN
- 45d SumNN
- # only
**Question Answering: Quiz Bowl Competition**

**QUESTION:**
He left unfinished a novel whose title character forges his father's signature to get out of school and avoids the draft by feigning desire to join. A more famous work by this author tells of the rise and fall of the composer Adrian Leverkühn. Another of his novels features the Jesuit Napthta and his opponent Settembrini, while his most famous work depicts the aging writer Gustav von Aschenbach. Name this German author of The Magic Mountain and Death in Venice.

**Iyyer et al. 2014: A Neural Network for Factoid Question Answering over Paragraphs**
Question Answering: Quiz Bowl Competition

• QUESTION:
He left unfinished a novel whose title character forges his father's signature to get out of school and avoids the draft by feigning desire to join. A more famous work by this author tells of the rise and fall of the composer Adrian Leverkühn. Another of his novels features the jesuit Naptha and his opponent Settembrini, while his most famous work depicts the aging writer Gustav von Aschenbach. Name this German author of The Magic Mountain and Death in Venice.

• ANSWER: Thomas Mann
Recursive Neural Networks

- Follow dependency structure

Students ride bikes at night
Ranking-type cost function

\[ J(\theta) = \sum_{i=1}^{N} \max\{0, 1 - s(n_i) + s(\bar{n}_i)\} \]

max \{ 0, 1 - C^T * S + I C^T * S \} 

Trained on Wikipedia and the QA corpus
Pushing Facts into Entity Vectors
Qanta Model Can Defeat Human Players

History: Model vs. Human

Won on some sub-problems
Literature Questions are Hard!

Not enough to look at the wikipedia entry and the question – you actually need to read the books!
Visual Grounding

- Idea: Map sentences and images into a joint space

Socher et al. 2013: Grounded Compositional Semantics for Finding and Describing Images with Sentences
• CNN trained on ImageNet (Le et al. 2013)
• RNN trained to give large inner products between sentence and image vectors:

\[ J(W_I, \theta) = \sum_{(i,j) \in \mathcal{P}} \sum_{c \in \mathcal{S} \setminus \mathcal{S}(i)} \max(0, \Delta - v_i^T y_j + v_i^T y_c) \]
Results

A gray convertible sports car is parked in front of the trees.
A close-up view of the headlights of a blue old-fashioned car.
Black shiny sports car parked on concrete driveway.
Five cows grazing on a patch of grass between two roadways.

A jockey rides a brown and white horse in a dirt corral.
A young woman is riding a Bay hose in a dirt riding-ring.
A white bird pushes a miniature teal shopping cart.
A person rides a brown horse.

A motocross bike with rider flying through the air.
White propeller plane parked in middle of grassy field.
The white jet with its landing gear down flies in the blue sky.
An elderly woman catches a ride on the back of the bicycle.

Demo: http://etcml.com:8080/
## Results

![Image](image.png)

<table>
<thead>
<tr>
<th>Describing Images</th>
<th>Mean Rank</th>
<th>Image Search</th>
<th>Mean Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>92.1</td>
<td>Random</td>
<td>52.1</td>
</tr>
<tr>
<td>Bag of Words</td>
<td>21.1</td>
<td>Bag of Words</td>
<td>14.6</td>
</tr>
<tr>
<td>CT—RNN</td>
<td>23.9</td>
<td>CT—RNN</td>
<td>16.1</td>
</tr>
<tr>
<td>Recurrent Neural Network</td>
<td>27.1</td>
<td>Recurrent Neural Network</td>
<td>19.2</td>
</tr>
<tr>
<td>Kernelized Canonical Correlation Analysis</td>
<td>18.0</td>
<td>Kernelized Canonical Correlation Analysis</td>
<td>15.9</td>
</tr>
<tr>
<td>DT—RNN</td>
<td><strong>16.9</strong></td>
<td>DT—RNN</td>
<td><strong>12.5</strong></td>
</tr>
</tbody>
</table>
Image – Sentence Generation (!)

- Several models came out simultaneously in 2015 that follow up
- Replace recursive neural network with LSTM and instead of only finding vectors they generate the description
- Mostly memorized training sequences (becomes similar again)

- Donahue et al. 2015: Long–term \( \rightarrow \) Recurrent Convolutional Networks for Visual Recognition and Description

Image – Sentence Generation (!)

"little girl is eating piece of cake."
"baseball player is throwing ball in game."
"woman is holding bunch of bananas."

"a young boy is holding a baseball bat."
"a cat is sitting on a couch with a remote control."
"a woman holding a teddy bear in front of a mirror."
Here come the Chat Bots!

- Greatest buzz in the VC community: Conversational Agents / Dialog Systems
- Microsoft, Facebook (M), Apple (Siri), Google, WeChat, and Slack are all in it with a wave of other startups: Operator, x.ai etc
- Platforms: Chatfuel, Howdy’s Botkit and Microsoft Bot Framework

Chat Bots Models

• Retrieval-based models (easier)
  • Use a repository of predefined responses and some kind of heuristic to pick an appropriate response based on the input and context.

• Generative models (harder)
  • Generate new responses from scratch, usually based on the “encoder-decoder” framework we studied
  • Current research is on the sequence to sequence techniques
Chat Bots Models

- **Short-Text Conversations (easier)**
  - create a single response to a single input
- **Long conversations (harder)**
  - go through multiple turns and need to keep track of what has been said. Customer support conversations are typically long conversational threads with multiple questions.
Chat Bots Models

- **Closed domain (easier)**
  - the space of possible inputs and outputs is somewhat limited because the system is trying to achieve a very specific goal. E.g., Technical Customer Support or Shopping Assistants

- **Open domain (harder)**
  - the user can take the conversation anywhere, e.g., conversations on social media sites like Twitter and Reddit
  - The infinite number of topics and the fact that a certain amount of world knowledge is required to create reasonable responses makes this a hard problem.
The Challenges

- **Incorporating Context**
  - Incorporate both *linguistic context* and *physical context*.
  - General Method: embed the conversation into a vector
    - Hard for long conversations
    - **Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models**
    - **Attention with Intention for a Neural Network Conversation Model**
The Challenges

• **Coherent Personality**
  • Agent should ideally produce consistent answers to semantically identical inputs: “How old are you?” “What is your age?”
  • Systems learn to generate linguistic plausible responses, but they are not trained to generate semantically consistent ones (usually trained on data from multiple users!)
• **A Persona-Based Neural Conversation Model**

<table>
<thead>
<tr>
<th>message</th>
<th>response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where were you born?</td>
<td>I was born in Canada.</td>
</tr>
<tr>
<td>Where are you from?</td>
<td>England, you?</td>
</tr>
<tr>
<td>Where did you grow up?</td>
<td>I grew up in Texas.</td>
</tr>
<tr>
<td>How old are you?</td>
<td>16 and you?</td>
</tr>
<tr>
<td>What’s your age?</td>
<td>18.</td>
</tr>
</tbody>
</table>
The Challenges

- **Evaluation**
  - Ideally - evaluate a conversational agent by measuring whether or not it is fulfilling its task, e.g. solve a customer support problem. But too expensive to get labels....
  - Common metrics such as BLEU that are used for Machine Translation and are based on text matching aren’t well suited because sensible responses can contain completely different words or phrases
The Challenges

- **Intention and Diversity**
  - Tend to produce generic responses like “That’s great!” or “I don’t know”
    - Early versions of Google’s Smart Reply tended to respond with “I love you” to almost anything.
  - Some early work on diversity:
    - Use Minimum Mutual Information (MMI) as an optimization objective that measures the mutual dependence between inputs and outputs instead of the usual optimization of log-likelihood of target T given source S
What is next?

“Most of the value of deep learning today is in narrow domains where you can get a lot of data. Here’s one example of something it cannot do: have a meaningful conversation. There are demos, and if you cherry-pick the conversation, it looks like it’s having a meaningful conversation, but if you actually try it yourself, it quickly goes off the rails.” (Andrew Ng)

• Many companies start off by outsourcing their conversations to human workers and promise that they can “automate” it once they’ve collected enough data. Might work well in closed domains.
• Learn to control your bot: “Microsoft is deleting its AI chatbot's incredibly racist tweets”
Getting ready to get your papers published!