Linguistic Essentials and NLP Applications

Besnik Fetahu
Today’s Lecture

• NLP Fundamentals
• Typical NLP tasks
• Word Representations — word2vec
• Recurrent Neural Networks — RNNs
• Bias in Language
• Research Papers Overview
NLP Fundamentals
Natural Language Processing

• NLP is the task of processing natural language in an automated manner

• Language is inherently difficult to automatically process and understand due to:
  • Ambiguity (syntactic, semantic)
  • Genre/Domain
  • Spatial, temporal context
  • Prior information (speaker background, common sense etc.)
• Language is filled with non-categorical phenomena:

• Language change are gradual and can be traced by analyzing the word frequency and its context:
  • “while” used as a noun to indicate time, now it is used as complementizer (subordinate clauses)
  • “gay” used to indicate predominantly happiness (emotional state), now used to indicate sexual orientation.

• Words can have multiple syntactic and semantic senses:
  • “bank” can refer to the river bank or financial institution etc.
  • “can” can be a verb or a noun

• Probabilistic approaches are best suitable for natural language understanding:
  • Incorporate priors (world priors, contextualized priors)
  • Incomplete information from a language utterance
Natural Language Processing

- **Homophone**: Same pronunciation, different meaning
  - Same pronunciation
- **Homograph**: Same spelling, different meaning
  - Same spelling
- **Heterograph**: Different spelling and meaning
  - Different spelling
- **Homonym**: Different meaning
  - Different pronunciation
- **Heteronym**: Different pronunciation and meaning
  - Different pronunciation
- **Synonym**: Different spelling and pronunciation
  - Different spelling
- **Words with different spelling, pronunciation and meaning**: Same meaning
NLP Tasks
Part of Speech Tagging

• POS tagging is the task of labeling each word in a sentence with its appropriate part of speech.

• 36 POS tags in Penn Treebank:

• Nouns, verbs, prepositions, adjectives etc.

```python
sentence = 'The quick brown fox jumped over the lazy dog.'

NLTK MaxEnt Tagger

tokens = nltk.word_tokenize(sentence)
pos_tags = nltk.pos_tag(tokens)
print str(pos_tags)

[(u'The', u'DT'), (u'quick', u'JJ'), (u'brown', u'NN'), (u'fox', u'NN'), (u'jumped', u'VBD'), (u'over', u'IN'), (u'the', u'DT'), (u'lazy', u'NN'), (u'dog', u'NN'), (u'.', u'.')]

Stanford CoreNLP CRF Tagger

print (nlp.pos_tag(sentence))

[(u'The', u'DT'), (u'quick', u'JJ'), (u'brown', u'NN'), (u'fox', u'NN'), (u'jumped', u'VBD'), (u'over', u'IN'), (u'the', u'DT'), (u'lazy', u'JJ'), (u'dog', u'NN'), (u'.', u'.')]
```
NER is the process of resolving words/surface forms into a predefined class of named entity categories (e.g. Person, Location, Organization):
Word sense disambiguation (WSD): determines the correct sense of a word given its context.

The robot that **can** recycle a **can** is useful for the environment.
Phrase structure parsing organizes syntax into constituents or brackets. In general, this involves nested trees.
Named Entity Disambiguation

- Named entity disambiguation (NED) is the task of resolving surface forms based on their context to entities from a reference database.
Word Representations
What is the meaning of a word?

- Traditional notion of word meaning in philosophy is determined by just the spelling of the word (symbolic representation). E.g. “CAT” represents the meaning of cat.

- This is insufficient representation of word meaning. There are word classes that express similar meaning, opposites, emotions, hierarchies etc.

- Determining the word meaning and resolving relations between words (e.g. relatedness, similarity, connotation etc.) is crucial to several NLP task: question answering, paraphrasing, summarization, … etc.
• **Lemmas and Senses:** A standard way to represent a word is through their lemmas.

• Lemmas represent the base form of a word from which we can derive other forms. This is a standard representation in dictionaries or other lexicons like WordNet.

• Senses represent the different meanings a word can take.

### Noun
- **S:** (n) mean, **mean value** (an average of n numbers computed by adding some function of the numbers and dividing by some function of n)

### Verb
- **S:** (v) mean, **intend** (mean or intend to express or convey) "You never understand what I mean!"; "what do his words intend?"
- **S:** (v) entail, **imply, mean** (have as a logical consequence) "The water shortage means that we have to stop taking long showers"
- **S:** (v) mean, **intend, signify, stand for** (denote or connote) "'maison' means 'house' in French"; "An example sentence would show what this word means"

### Adjective
- **S:** (adj) **average, mean** (approximating the statistical norm or average or expected value) "the average income in New England is below that of the nation"; "of average height for his age"; "the mean annual rainfall"
- **S:** (adj) **hateful, mean** (characterized by malice) "a hateful thing to do"; "in a mean mood"
• **Synonyms:** The relationship between words and their senses that have identical senses are called synonyms.

  • couch/sofa   car/automobile   PC/computer

  • Words are considered to be synonyms if they can be used in the same sentence without necessarily changing the *truth conditions* in the sentence. In this case it is said that two words have the same *propositional meaning*.

  • In practice, synonyms are *not identical*. The *principle of contrast* states that the linguistic difference among the different synonyms can be in terms of *genre*, or other context specific difference.
Antonyms are the exact opposite of synonyms. They represent the relationship between words or senses that have the opposite meaning.

- long/short  fast/slow  dark/light

This relationship holds w.r.t some scaling or direct comparison measure between two words.

Reversives are a subgroup of antonyms, they express the opposite meanings of words described in terms of some movement/direction scale.

- up/down  rise/fall
Wittgenstein suggests that “the meaning of a word is its use in language”.

Distributional hypothesis “words that occur in the same context have similar meaning” (Harris, 1954)

“You shall know a word by the company it keeps” (Firth, 1957)

These definitions form what is known vector semantics or word representations.

Vector semantics combine two intuitions: (i) the distributionalist intuition (that a word is defined by its context), and the vector intuition of Osgood et al., where the word can be defined across multiple dimension (aspects).
Suppose you see the following sentences:

- Ong choi is delicious sauteed with garlic.
- Ong choi is superb over rice.
- Ong choi leaves with salty sauces.

And suppose you've seen also the following sentence:

- … spinach sauteed with garlic over rice.

Chard stems and leaves are delicious.
Collard greens and other salty leafy greens.

Conclusion:

- Ongchoi is a leafy green like spinach, chard, or collard greens.
word2vec — skip gram
Maximize the probability of the word \( w_t \) (for “target”) given the previous words \( h \) (for “history”) in terms of a softmax function:

Score: computes word/history compatibility (typically dot-product)

Softmax: squashes a vector of real values to a vector of real values in the range (0,1) that add up to 1.
Contrary to PPI and LSA embeddings, the neural based word embeddings (skipgram, and cbow) are prediction based embeddings.

Learn word embeddings as a process of word prediction (context or center word).

Fast and easy to train. The training data consists of sample word sequences of a pre-determined context window.

Two of the most well known ways to train neural word embeddings are:

- **cbow**: given some context predict the center word or the next word (similar to language models)

- **skip-gram**: given a center word, predict the context word with which the word co-occurs.
Word2vec Word Embeddings

Noise classifier

Hidden layer

Projection layer

\[ \sum g(\text{embeddings}) \]

the cat sits on the mat
Word2vec Word Embeddings

- Trains a classifier on a binary classification task:
  - skip-gram: Is a context word likely to appear with a center word?
  - cbow: Is a center word likely to appear given some context?
- No need for hand labelled data. Use standard natural language utterances from the Web.
- The learning task itself is not important, but we’ll use the learned classifier weights as word embeddings.
• Training sentence:
  ★ “… lemon, a tablespoon of apricot jam a pinch …”

• skip-gram with context window of +/- 2 words:
  ★ “… lemon, a tablespoon of apricot jam a pinch …”

• cbow with a context window of 4 words:
  ★ “… lemon, a tablespoon of apricot jam a pinch …”
Word2vec Skip-Gram

- Treat the target word and a neighboring context word as positive examples.

- Randomly sample other words from the lexicon to get negative samples (skip-gram with negative sampling or SGNS)

- Use logistic regression to train a classifier that distinguishes between the positive and negative cases.

- Use the learned weights as word embeddings.
Given a tuple \((t, c)\) = target, context words
- \((\text{apricot}, \text{jam})\)
- \((\text{apricot}, \text{aardvark})\)

Return the probability that \(c\) is a real context word:

\[
P( + | t, c) \quad P( - | t, c) = 1 - P( + | t, c)
\]
How can we compute $P( + \mid t, c) \quad P(w_{t+c} \mid w_t)$?

Intuition:
- Words are likely to appear near similar words
- Model similarity with dot product!
  - *Dot-product does not give us the probability!!*

We use a logistic (sigmoid) function to turn the dot product between the embeddings of the words into a probability.

$$P( + \mid t, c) = \frac{1}{1 + e^{-t \cdot c}} \quad P( - \mid t, c) = \frac{e^{-t \cdot c}}{1 + e^{-t \cdot c}}$$
Word2Vec Skip-Gram

Training procedure:
- For each positive example create k negative samples
- Use noise words (randomly select from a lexicon, usually from 5–20 words per positive pair)
Word2Vec Skip-Gram

The diagram illustrates the concept of Word2Vec Skip-Gram, which is a method for training word embeddings. The figure shows how the similarity between words can be increased or decreased based on their context. For example, the similarity between "apricot" and "jam" is increased ($w_j \cdot c_k$), while the similarity between "apricot" and "aardvark" is decreased ($w_j \cdot c_n$).
Recurrent Neural Networks — RNN
RNNs — Introduction

• Modell and process sequential information (language modeling tasks, gene sequencing etc.).

• RNNs are called “recurrent” because they apply the same processing to each element in a sequence.

• The output for a given element in the sequence is dependent on the previous observations.

• RNNs through its parameters contains a memory of the previous processed elements, independent on the length of the sequence.

• The core intuition lies in preserving previous information in order to predict the next state.
RNN — Models

- **one to one**
- **one to many**
- **many to one**
- **many to many**

Diagram descriptions:
- **one to one**: Single input and output.
- **one to many**: Single input to multiple outputs.
- **many to one**: Multiple inputs to a single output.
- **many to many**: Multiple inputs to multiple outputs.
RNN — Many to Many

- Language Modeling
- Machine Translation
RNN Architecture
RNN — Architecture

**Input:** e.g. one-hot vector

**Model:** forward pass and backward pass

**Output:** discrete output (e.g. POS tags, words, chars)
RNN — Architecture

Recurrent Process

© http://colah.github.io/posts/2015-08-Understanding-LSTMs/
RNN — Model Parameters

- **RNN**
  - **U**: input to hidden
  - **W**: hidden to hidden
  - **V**: hidden to output

- **b, c**: bias vectors
RNN — Model Parameters

Model parameters are shared!!
Forward Pass
RNN — Forward Pass

\[ h^{(t)} = f(b + Wh^{(t-1)} + Ux^{(t)}) \]

**hidden state activations**

\[ o^{(t)} = \text{softmax}(c + Vh^{(t)}) \]

**output state**

- \( f \) is usually a **tanh** or **sigmoid** function
- (binary or multinomial)
RNN — Forward Pass

All parameters are shared ($W, U, V$)
Backward Pass
• The loss function is the **cross entropy** function.

• Use stochastic gradient descent for a batch of sequences to compute the gradient and update the parameters.

• Computing for each sequence is very costly.

• The algorithm for updating the parameters $W, U, V, b, c$ is called **back-propagation through time** (BPTT)
Back-propagating the gradient from the different timestamps in our sequence for $W$?
Language Bias

“Neural Based Statement Classification for Biased Language”. Christoph Hube, and Besnik Fetahu. WSDM 2019; Melbourne, Australia (to appear).

"Detecting Biased Statements in Wikipedia". Christoph Hube, and Besnik Fetahu. WWW (Companion Volume) 2018; Lyon, France.
• What if claims make use of highly subjective language?

• What if claims make use of highly one-sided language?

• What are the language indicators of bias?

• How can we flag such statements based solely on the used language?
Where does bias occur?

- Newspapers, (online) news sources
- Social media, blogs, forums
- Encyclopedias

- Typical topics:
  - politics, cultures, genders, history, other controversial topics
  - Is bias desirable?
    - Obvious opinion or neutral information source? “Common sense”?  
    - Echo chambers
What are the types of bias?

• Typical types of bias:
  
  • **Phrasing bias**: opinionated wording, use of inflammatory or partial words or phrases
  
  • **Selection bias/coverage bias**: selecting facts that support a specific POV, omitting other facts
  
  • **Focus bias**: Positioning specific statements at central parts in the article (e.g. headlines)
Linguistic Cues for Phrasing Bias
Forms of Linguistic Bias

a. Usually, smaller cottage-style houses have been demolished to make way for these McMansions.
b. Usually, smaller cottage-style houses have been demolished to make way for these homes.

**Framing bias:** language that is biased towards some specific point of view.

a. Kuypers **claimed** that the mainstream press in America tends to favor liberal viewpoints.
b. Kuypers **stated** that the mainstream press in America tends to favor liberal viewpoints.

**Epistemological bias:** linguistic features that try to weaken the believability of a statement.

Epistemological Bias

Factive Verbs: presuppose the truth of their complement clause

Epistemological Bias

Entailment: are directional relations that hold whenever the truth of one word or phrase follows from another, e.g., *murder* entails *kill*

*murder vs. kill* entails killing in an unlawful, premeditated way

Epistemological Bias

Hedges: reduce one’s commitment to the truth of a proposition

Assertive verbs: The truth of the proposition is not presupposed, but its level of certainty depends on the asserting verb

Say vs. claim: neutral vs. casting doubt in a proposition

Framing Bias

Subjective: are adjectives or adverbs that add (subjective) force to the meaning of a phrase or proposition

Framing Bias

One sided: reflect only one of the sides of a contentious issue. They often belong to controversial subjects (e.g., religion, terrorism, etc.)
• Language Bias depends on the context
  
  (a) Andrew James Breitbart was one of the most outspoken, fearless conservative journalists in America.

  (b) The Labour Party in the United Kingdom put together a highly successful set of policies based on encouraging the market economy, while promoting the involvement of private industry in delivering public services.

  (c) An abortion is the murder of a human baby embryo or fetus from the uterus, resulting in or caused by its death.

  (d) Sanders shocked his fellow liberals by putting up a Soviet Union flag in his Senate office.

  (e) This may be a result of the fact that the public had unsurprisingly lost support for the President and his policies.

  (f) The Blair government had promised a referendum on whether Britain should sign the Constitution, but refused popular demands that it carry out its promise.

• Need for a context-sensitive approach: Use RNN with GRUs for classifying statements if they contain biased language.
How can we capture the language bias context?
Different types of statement representations in the RNN model:

- **Word representations**: Word embeddings (GloVe)
- **POS tags**: Part-of-Speech tags, able to capture stylistic linguistic features
- **LIWC word functions**: psychological and sociological functions of words, most descriptive function per word
- Combine different statement representations
Not all words are important for detecting language bias

• Use attention weights to focus on specific parts of a statement

  • “The **public agrees** that it is **the number one** country in the world.”

• Use attention mechanism:

  • Global attention

  • Hierarchical attention
RNN with Hierarchical Attention

Compute importances for different statement representations separately
• POV tags in revision comments

• Extracting single modified or deleted statements with POV tag

• Similar to Recasens et al. (ACL, 2016)
  • 280k candidates for biased statements

• False positives:
  • POV tag does not refer to statement, e.g.: “Can someone explain to me what is POV about this article?”
  • Different types of bias (e.g. focus and selection bias)
  • POV tag by a single editor
Crowdsourced ground-truth dataset consists of:

- 1843 statements labeled as containing phrasing bias (~37%)
- 3109 statements labeled as neutral (~62%)

Dataset of varying difficulty:

- **CW-HARD**: Statements labeled as “neutral wording” in our dataset still belong to the subset of POV tagged statements classification task hard;
- **Featured**: extract *neutral statements* from featured articles
- **Type-balanced**: Align article type distributions (use a similar distribution based on the article type from where the revisions come from *Persons*, *Locations* etc.)
## Language Bias Evaluation Results

<table>
<thead>
<tr>
<th>featured</th>
<th>Acc</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>0.646</td>
<td>0.65</td>
<td>0.632</td>
</tr>
<tr>
<td>B2</td>
<td>0.702</td>
<td>0.703</td>
<td>0.7</td>
</tr>
<tr>
<td>RNN$^w$</td>
<td>0.776</td>
<td>0.788</td>
<td>0.78</td>
</tr>
<tr>
<td>RNN$^{wp}$</td>
<td>0.789</td>
<td>0.843</td>
<td>0.717</td>
</tr>
<tr>
<td>RNN$^{wl}$</td>
<td>0.794</td>
<td>0.851</td>
<td>0.717</td>
</tr>
<tr>
<td>RNN$^{wpl}$</td>
<td>0.771</td>
<td>0.803</td>
<td>0.738</td>
</tr>
<tr>
<td>RNN$_a^w$</td>
<td>0.795</td>
<td>0.866</td>
<td>0.691</td>
</tr>
<tr>
<td>RNN$_a^{wp}$</td>
<td>0.818</td>
<td>0.892</td>
<td>0.715</td>
</tr>
<tr>
<td>RNN$_a^{wl}$</td>
<td>0.8</td>
<td>0.809</td>
<td>0.809</td>
</tr>
<tr>
<td>RNN$_a^{wpl}$</td>
<td>0.801</td>
<td>0.86</td>
<td>0.723</td>
</tr>
<tr>
<td>RNN$_h^{wp}$</td>
<td>0.803</td>
<td>0.899</td>
<td>0.649</td>
</tr>
<tr>
<td>RNN$_h^{wl}$</td>
<td>0.819</td>
<td>0.917</td>
<td>0.668</td>
</tr>
<tr>
<td>RNN$_h^{wpl}$</td>
<td>0.812</td>
<td>0.872</td>
<td>0.733</td>
</tr>
</tbody>
</table>
## Language Bias Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th>type-balanced</th>
<th>featured</th>
<th>cw-hard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F1</strong></td>
<td><strong>F1</strong></td>
<td><strong>F1</strong></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>0.663</td>
<td>0.641</td>
<td>0.616</td>
</tr>
<tr>
<td>B2</td>
<td>0.708</td>
<td>0.7</td>
<td>0.643</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;w&lt;/sup&gt;</td>
<td>0.7</td>
<td></td>
<td>0.668</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;wp&lt;/sup&gt;</td>
<td>0.77</td>
<td></td>
<td>0.602</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;wl&lt;/sup&gt;</td>
<td>0.78</td>
<td></td>
<td>0.681</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;wpl&lt;/sup&gt;</td>
<td>0.766</td>
<td>0.769</td>
<td>0.654</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;a,w&lt;/sup&gt;</td>
<td>0.776</td>
<td>0.769</td>
<td>0.699</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;a,wp&lt;/sup&gt;</td>
<td>0.797</td>
<td>0.794</td>
<td>0.678</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;a,wl&lt;/sup&gt;</td>
<td>0.8</td>
<td>0.809</td>
<td>0.705</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;a,wpl&lt;/sup&gt;</td>
<td>0.778</td>
<td>0.785</td>
<td>0.7</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;h,wp&lt;/sup&gt;</td>
<td>0.768</td>
<td>0.754</td>
<td>0.666</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;h,wl&lt;/sup&gt;</td>
<td>0.764</td>
<td>0.773</td>
<td>0.702</td>
</tr>
<tr>
<td>RNN&lt;sup&gt;h,wpl&lt;/sup&gt;</td>
<td>0.786</td>
<td>0.797</td>
<td>0.69</td>
</tr>
</tbody>
</table>

**B1:** $\Delta F1 = 7\%$

**B2:** $\Delta F1 = 10\%$
Paper Outline
Word embedding models such as GloVe rely on co-occurrence statistics to learn vector representations of word meaning. While we may similarly expect that co-occurrence statistics can be used to capture rich information about the relationships between different words, existing approaches for modeling such relationships are based on manipulating pre-trained word vectors. In this paper, we introduce a novel method which directly learns relation vectors from co-occurrence statistics. To this end, we first introduce a variant of GloVe, in which there is an explicit connection between word vectors and PMI weighted co-occurrence vectors. We then show how relation vectors can be naturally embedded into the resulting vector space.
Cross-lingual Entity Linking (XEL) aims to ground entity mentions written in any language to an English Knowledge Base (KB), such as Wikipedia. XEL for most languages is challenging, owing to limited availability of resources as supervision. We address this challenge by developing the first XEL approach that combines supervision from multiple languages jointly. This enables our approach to: (a) augment the limited supervision in the target language with additional supervision from a high-resource language (like English), and (b) train a single entity linking model for multiple languages, improving upon individually trained models for each language. Extensive evaluation on three benchmark datasets across 8 languages shows that our approach significantly improves over the current state-of-the-art. We also provide analyses in two limited resource settings: (a) zero-shot setting, when no supervision in the target language is available, and in (b) low-resource setting, when some supervision in the target language is available. Our analysis provides insights into the limitations of zero-shot XEL approaches in realistic scenarios, and shows the value of joint supervision in low-resource settings.
Many tasks in natural language processing involve comparing two sentences to compute some notion of relevance, entailment, or similarity. Typically, this comparison is done either at the word level or at the sentence level, with no attempt to leverage the inherent structure of the sentence. When sentence structure is used for comparison, it is obtained during a non-differentiable preprocessing step, leading to propagation of errors. We introduce a model of structured alignments between sentences, showing how to compare two sentences by matching their latent structures. Using a structured attention mechanism, our model matches candidate spans in the first sentence to candidate spans in the second sentence, simultaneously discovering the tree structure of each sentence. Our model is fully differentiable and trained only on the matching objective. We evaluate this model on two tasks, entailment detection and answer sentence selection, and find that modeling latent tree structures results in superior performance. Analysis of the learned sentence structures shows they can reflect some syntactic phenomena.
Comprehending procedural text, e.g., a paragraph describing photosynthesis, requires modeling actions and the state changes they produce, so that questions about entities at different timepoints can be answered. Although several recent systems have shown impressive progress in this task, their predictions can be globally inconsistent or highly improbable. In this paper, we show how the predicted effects of actions in the context of a paragraph can be improved in two ways: (1) by incorporating global, commonsense constraints (e.g., a non-existent entity cannot be destroyed), and (2) by biasing reading with preferences from large-scale corpora (e.g., trees rarely move). Unlike earlier methods, we treat the problem as a neural structured prediction task, allowing hard and soft constraints to steer the model away from unlikely predictions. We show that the new model significantly outperforms earlier systems on a benchmark dataset for procedural text comprehension (+8% relative gain), and that it also avoids some of the nonsensical predictions that earlier systems make.
Thank you!
Questions?