Foundations of Artificial Intelligence

15. Natural Language Processing
Understand, interpret, manipulate, generate human language
(text and audio)

Joschka Boedecker and Wolfram Burgard and
Frank Hutter and Bernhard Nebel and Michael Tangermann

Albert-Ludwigs-Universität Freiburg

July 17, 2019
1 Motivation, NLP Tasks

2 Learning Representations

3 Sequence-to-Sequence Deep Learning
Example: Automated Online Assistant

Source: Wikicommmons/Bemidji State University
Lecture Overview

1. Motivation, NLP Tasks
2. Learning Representations
3. Sequence-to-Sequence Deep Learning
The language of humans is represented as text or audio data. The field of NLP creates interfaces between human language and computers.

Goal: automatic processing of large amounts of human language data.
Examples of NLP Tasks and Applications

- word stemming
- word segmentation, sentence segmentation
- text classification
- sentiment analysis (polarity, emotions, ..)
- topic recognition
- automatic summarization
- machine translation (text-to-text)
- speaker identification
- speech segmentation (into sentences, words)
- speech recognition (i.e. speech-to-text)
- natural language understanding
- text-to-speech
- text and spoken dialog systems (chatbots)
From Rules to Probabilistic Models to Machine Learning

Part-of-Speech Tagging:

I can light a fire and you can open a can of beans. Now the can is open and we can eat in the light of the fire.

I/PRP can/MD light/VB a/DT fire/NN and/CC you/PRP can/MD open/VB a/DT can/NN of/IN beans/NNS ./ Now/RB the/DT can/NN is/VBZ open/JJ and/CC we/PRP can/MD eat/VB in/IN the/DT light/NN of/IN the/DT fire/NN ./

Sources: Slide by Torbjörn Lager; (Anthony, 2013)

Traditional rule-based approaches and (to a lesser degree) probabilistic NLP models faced limitations, as

- human don’t stick to rules, commit errors.
- language evolves: rules are neither strict nor fixed.
- labels (e.g. tagged text or audio) were required.

Machine translation was extremely challenging due to shortage of multilingual textual corpora for model training.
Machine learning entering the NLP field:

- Since late 1980’s: increased data availability (WWW)
- Since 2010’s: huge data, computing power → unsupervised representation learning, deep architectures for many NLP tasks.
Lecture Overview

1 Motivation, NLP Tasks

2 Learning Representations

3 Sequence-to-Sequence Deep Learning
A word embedding $W$ is a function

$$W : \text{words} \rightarrow \mathbb{R}^n$$

which maps words of some language to a high-dimensional vector space (e.g. 200 dimensions).

Examples:

$$W(\text{"cat"}) = (0.2, -0.4, 0.7, \ldots)$$
$$W(\text{"mat"}) = (0.0, 0.6, -0.1, \ldots)$$

Mapping function $W$ should be realized by a look-up table or by a **neural network** such that:

- representations in $\mathbb{R}^n$ of related words have a short distance
- representations in $\mathbb{R}^n$ of unrelated words have a large distance

How can we learn a good representation / word embedding function $W$?
A word embedding function $W$ can be trained using different tasks, that require the network to discriminate related from unrelated words.

Can you think of such a training task? Please discuss with your neighbors!
A word embedding function $W$ can be trained using different tasks, that require the network to discriminate related from unrelated words.

Can you think of such a training task? Please discuss with your neighbors!
A word embedding function $W$ can be trained using different tasks, that require the network to discriminate related from unrelated words.

Example task: predict, if a 5-gram (sequence of five words) is valid or not. Training data contains valid and slightly modified, invalid 5-grams:

$R(W("cat"), W("sat"), W("on"), W("the"), W("mat")) = 1$

$R(W("cat"), W("sat"), W("song"), W("the"), W("mat")) = 0$

... 

Train the combination of embedding function $W$ and classification module $R$:

While we may not be interested in the trained module $R$, the learned word embedding $W$ is very valuable!
Visualizing the Word Embedding

Let’s look at a projection from $\mathbb{R}^n \rightarrow \mathbb{R}^2$ obtained by tSNE:
Visualizing the Word Embedding

Let’s look at a projection from $\mathbb{R}^n \rightarrow \mathbb{R}^2$ obtained by tSNE:

t-SNE visualizations of word embeddings. Left: Number Region; Right: Jobs Region. From Turian et al. (2010)
Sanity Check: Word Similarities in $\mathbb{R}^n$?

<table>
<thead>
<tr>
<th>FRANCE</th>
<th>JESUS</th>
<th>XBOX</th>
<th>REDDISH</th>
<th>SCRATCHED</th>
<th>MEGABITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUSTRIA</td>
<td>GOD</td>
<td>AMIGA</td>
<td>GREENISH</td>
<td>NAILED</td>
<td>OCTETS</td>
</tr>
<tr>
<td>BELGIUM</td>
<td>SATI</td>
<td>PLAYSTATION</td>
<td>BLUISH</td>
<td>SMASHED</td>
<td>MB/S</td>
</tr>
<tr>
<td>GERMANY</td>
<td>CHRIST</td>
<td>MSX</td>
<td>PINKISH</td>
<td>PUNCHED</td>
<td>BIT/S</td>
</tr>
<tr>
<td>ITALY</td>
<td>SATAN</td>
<td>IPOD</td>
<td>PURPLISH</td>
<td>POPPED</td>
<td>BAUD</td>
</tr>
<tr>
<td>GREECE</td>
<td>KALI</td>
<td>SEGA</td>
<td>BROWNISH</td>
<td>CRIMPED</td>
<td>CARATS</td>
</tr>
<tr>
<td>SWEDEN</td>
<td>INDRA</td>
<td>PSNUMBER</td>
<td>GREYISH</td>
<td>SCRAPED</td>
<td>KBIT/S</td>
</tr>
<tr>
<td>NORWAY</td>
<td>VISHNU</td>
<td>HD</td>
<td>GRAYISH</td>
<td>SCREWED</td>
<td>MEGAHERTZ</td>
</tr>
<tr>
<td>EUROPE</td>
<td>ANANDA</td>
<td>DREAMCAST</td>
<td>WHITISH</td>
<td>SECTIONED</td>
<td>MEGAPIXELS</td>
</tr>
<tr>
<td>HUNGARY</td>
<td>PARVATI</td>
<td>GEFORCE</td>
<td>SILVERY</td>
<td>SLASHED</td>
<td>GBIT/S</td>
</tr>
<tr>
<td>SWITZERLAND</td>
<td>GRACE</td>
<td>CAPCOM</td>
<td>YELLOWISH</td>
<td>RIPPED</td>
<td>AMPERES</td>
</tr>
</tbody>
</table>

What words have embeddings closest to a given word? From Collobert *et al.* (2011)
Embedding allows to work not only with synonyms, but also with other words of the same category:

- "the cat is black" → "the cat is white"
- "in the zoo I saw an elephant" → "in the zoo I saw a lion"

In the embedding space, systematic shifts can be observed for analogies:

\[ w(\text{man}) - w(\text{woman}) \]

From Mikolov et al. (2013a)

The embedding space may provide dimensions for gender, singular-plural etc.
### Observed Relationship Pairs in the Learned Embedding $W$

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>France - Paris</td>
<td><strong>Italy</strong>: Rome</td>
<td><strong>Japan</strong>: Tokyo</td>
<td><strong>Florida</strong>: Tallahassee</td>
</tr>
<tr>
<td>big - bigger</td>
<td>small: larger</td>
<td>cold: colder</td>
<td>quick: quicker</td>
</tr>
<tr>
<td>Miami - Florida</td>
<td>Baltimore: Maryland</td>
<td>Dallas: Texas</td>
<td>Kona: Hawaii</td>
</tr>
<tr>
<td>Einstein - scientist</td>
<td>Messi: midfielder</td>
<td>Mozart: violinist</td>
<td>Picasso: painter</td>
</tr>
<tr>
<td>Sarkozy - France</td>
<td>Berlusconi: Italy</td>
<td>Merkel: Germany</td>
<td>Koizumi: Japan</td>
</tr>
<tr>
<td>copper - Cu</td>
<td>zinc: Zn</td>
<td>gold: Au</td>
<td>uranium: plutonium</td>
</tr>
<tr>
<td>Berlusconi - Silvio</td>
<td></td>
<td>Putin: Medvedev</td>
<td>Obama: Barack</td>
</tr>
<tr>
<td>Microsoft - Windows</td>
<td></td>
<td>IBM: Linux</td>
<td>Apple: iPhone</td>
</tr>
<tr>
<td>Microsoft - Ballmer</td>
<td></td>
<td>IBM: McNealy</td>
<td>Apple: Jobs</td>
</tr>
<tr>
<td>Japan - sushi</td>
<td></td>
<td>France: tapas</td>
<td>USA: pizza</td>
</tr>
</tbody>
</table>

Relationship pairs in a word embedding. From Mikolov *et al.* (2013b).
Various embedding models / strategies have been proposed:

- Word2vec (Tomas Mikolov et al., 2013)
- GloVe (Pennington et al., 2014)
- fastText library (released by Facebook by group around Tomas Mikolov)
- ELMo (Matthew Peters et al., 2018)
- ULMFit (by fast.ai founder Jeremy Howard and Sebastian Ruder)
- BERT (by Google)
- ...

(Pre-trained models are available for download)
Word Embeddings: the Secret Sauce for NLP Projects

Shared representations — re-use a pre-trained embedding for other tasks!

Using ELMo embeddings improved six state-of-the-art NLP models for:

- Question answering
- Textual entailment (inference)
- Semantic role labeling (”Who did what to whom?”)
- Coreference resolution (clustering mentions of the same entity)
- Sentiment analysis
- Named entity extraction
Can Neural Representation Learning Support Machine Translation?

Can you think of a training strategy to translate from Mandarin to English and back? Please discuss with your neighbors!
Can Neural Representation Learning Support **Machine Translation**?

Can you think of a training strategy to translate from Mandarin to English and back? Please discuss with your neighbors!
Bilingual Word Embedding

Idea: train two embeddings in parallel such, that corresponding words are projected to close-by positions in the word space.
Let’s again look at a tSNE projection $\mathbb{R}^n \rightarrow \mathbb{R}^2$:

$t$-SNE visualization of the bilingual word embedding. Green is Chinese, Yellow is English.  
(Socher et al. (2013a))
Lecture Overview

1. Motivation, NLP Tasks

2. Learning Representations

3. Sequence-to-Sequence Deep Learning
So far, the network has learned to deal with a **fixed number of input words** only.
So far, the network has learned to deal with a fixed number of input words only.

Limitation can be overcome by adding association modules, which can combine two word and phrase representations and merge them.
So far, the network has learned to deal with a **fixed number of input words** only.

Limitation can be overcome by adding **association modules**, which can combine two word and phrase representations and merge them.

Using associations, whole sentences can be represented!

(From Bottou (2011))
Conceptually, we could now use this concept to find the embedding of a word or sentence of the source language and look up the closest embedding of the target language.

What is missing to realize a translation?
For translations, we also need disassociation modules! (encoder — decoder principle)

(From Bottou (2011))
Sequence-to-Sequence Neural Machine Translation

Ground-breaking new approach by Bahdanau, Cho and Bengio (2014 ArXiv, 2015 ICML)

- Shift through the input word sequence
- Learn to encode and to decode using recurrent neural networks (RNN)
- Learn to align input and output word sequences
- Take context into account by learning the importance of neighboring words \(\rightarrow\) **attention mechanism**.

Credits: (Olah & Carter, 2016) have adapted this figure based on (Bahdanau et al., 2014)
Sequence-to-Sequence Neural Machine Translation

Ground-breaking new approach by Bahdanau, Cho and Bengio (2014 ArXiv, 2015 ICML)

- Shift through the input word sequence
- Learn to encode and to decode using recurrent neural networks (RNN)
- Learn to align input and output word sequences
- Take context into account by learning the importance of neighboring words → **attention mechanism**.

Credits: (Olah & Carter, 2016) have adapted this figure based on (Bahdanau et al., 2014)
Similar principle, but voice/speech input

Credits: (Olah & Carter, 2016) have adapted this figure based on (Chan et al., 2015)
Success Story of Attention-based Neural Machine Translation

Neural machine translation requires big data sets but has advantages:

- Overall model can be learned end-to-end
- No need to integrate modules for feature extraction, database, grammar rules etc. in a complicated system
Natural language processing spans a wide range of problems and applications.

NLP is a rapidly growing field due to availability of huge data sets.

NLP techniques are part of many products already.

Field is moving more and more to neural networks, which provide NLP building blocks like end-to-end learning, representation learning, sequence-to-sequence, ...