BETTER WORD REPRESENTATIONS WITH RECURSIVE NEURAL NETWORKS FOR MORPHOLOGY

MATEUSZ ZAN
S1113967
OUTLINE

• Key points of interest
• Motivation
• Possible applications
• Methodology (2 different approaches)
• Experiment (Word Similarity Task)
• Results and Evaluation
KEY POINTS OF INTEREST

• Morphology - study of patterns of word formation for a natural language

• Recurrent Neural Networks- statistical learning algorithm used in many NLP tasks

• Using RNNs, we aim to model morphology- be able to build better word representations for complex words, gradually building more complex words using morphemes
MOTIVATION

- 'formal'
- 'in-formal-ity'
POSSIBLE APPLICATIONS

- POS Tagging - certain suffixes correspond to different parts of speech
- Parsing (Morphological parsing)
- Semantic role labelling - breaking up words into smaller parts can help with labelling semantics
• Quick recap of neural networks!

• In this problem, we use a variant of Neural Network: Morphological Recursive Neural Network

• The Morphological Recursive Neural Network works on morpheme level, rather than on the word level
RNN MODEL

- Morphemes encoded as vectors (dimension d) in an embedding matrix $\mathbf{W}_e$ (of size $d \times |M|$, where $M$ is set of all morphemes)

- Words are built gradually, by combing morphological parts. Parent vector $p$ is constructed by combining stem vector with affix vector

- We have a model $\theta = \{\mathbf{W}_e, \mathbf{W}_m, \mathbf{b}_m\}$ and we want to learn the parameters

\[ p = f(\mathbf{W}_m [\mathbf{x}_{stem}; \mathbf{x}_{affix}] + \mathbf{b}_m) \]
Two different approaches of the Neural Network implementation considered:

- Context-insensitive Morphological RNN
- Context-sensitive Morphological RNN

The difference between the two is that the latter considers the contextual information (the other words in a sentence, etc.), while the other one doesn’t use that information.
CONTEXT-INSENSITIVE MODEL

• Model considers how words can be constructed simply from morphemic representation

• Given the reference words, the goal is to construct new words to match the reference as closely as possible

• Structure is the same as a basic RNN model, explained above

• For learning, a cost function $s$ is defined measuring the Euclidean distance between output and reference vector

• Objective function we try to minimise is the sum of cost for $N$ words:

$$J(\theta) = \sum_{i=1}^{N} s(x_i) + \frac{\lambda}{2} \| \theta \|_2^2$$
CONTEXT-SENSITIVE MODEL

• Tries to address limitations of the previous model by considering the context in which the word appears

• 2 layers- MorphoRNN and word-based neural language model

• n-grams scored using formula

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

• Objective function to optimize parameters

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

• Model parameters are \( \theta = \{W_e; W_m; b_m; W; b;\} \).
PARAMETER OPTIMISATION (LEARNING)

- Algorithm considers two passes: forward-pass and backward-pass
- For the latter, we are interested in minimising objective function, to optimise parameters (back-propagation)

\[
\frac{\partial J(\theta)}{\partial \theta} = \sum_{i=1}^{N} \frac{\partial s(x_i)}{\partial \theta} + \lambda \theta
\]

\[
\frac{\partial J(\theta)}{\partial \theta} = \sum_{i:1-s(n_i)+s(\bar{n}_i)>0} -\frac{\partial s(n_i)}{\partial \theta} + \frac{\partial s(\bar{n}_i)}{\partial \theta}
\]
• Morfessor- morphological segmentation toolkit

• For this investigation, we assume input of form pre*stm suf*

• Words in data are split using the toolkit, and affixes are stored, presented in a table

<table>
<thead>
<tr>
<th>Prefixes</th>
<th>Suffixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 al all anti</td>
<td>able al ally american ance</td>
</tr>
<tr>
<td>counter cross</td>
<td>ated ate ation backed bank</td>
</tr>
<tr>
<td>de dis electro</td>
<td>based born controlled d</td>
</tr>
<tr>
<td>end ex first</td>
<td>dale down ed en er es field</td>
</tr>
<tr>
<td>five focus four</td>
<td>ford free ful general head</td>
</tr>
<tr>
<td>high hyper ill</td>
<td>ia ian ible ic in ing isation</td>
</tr>
<tr>
<td>jan jean long</td>
<td>ise ised ish ism ist ity ive</td>
</tr>
<tr>
<td>low market mc</td>
<td>ization ize ized izing land</td>
</tr>
<tr>
<td>micro mid multi</td>
<td>led less ling listed ly made</td>
</tr>
<tr>
<td>neuro newly no</td>
<td>making man ment ness off</td>
</tr>
<tr>
<td>non off one</td>
<td>on out owned related s ship</td>
</tr>
<tr>
<td>over post pre pro</td>
<td>shire style ton town up us</td>
</tr>
<tr>
<td>re second self</td>
<td>ville wood</td>
</tr>
<tr>
<td>semi seven short</td>
<td></td>
</tr>
<tr>
<td>six state sub</td>
<td></td>
</tr>
<tr>
<td>super third</td>
<td></td>
</tr>
<tr>
<td>three top trans</td>
<td></td>
</tr>
<tr>
<td>two un under uni</td>
<td></td>
</tr>
<tr>
<td>well</td>
<td></td>
</tr>
</tbody>
</table>
EXPERIMENTAL SETUP

- Two embeddings (C&W- Collobert, HSMN- Huang)
- Various datasets: WS353, MC, RG, SCWS*, RW- various datasets to avoid overfitting
- Rare Word dataset - RW

<table>
<thead>
<tr>
<th></th>
<th>All words</th>
<th>Complex words</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS353</td>
<td>0</td>
<td>0 / 9 / 87 / 341</td>
</tr>
<tr>
<td>MC</td>
<td>0</td>
<td>0 / 1 / 17 / 21</td>
</tr>
<tr>
<td>RG</td>
<td>0</td>
<td>0 / 4 / 22 / 22</td>
</tr>
<tr>
<td>SCWS*</td>
<td>26</td>
<td>2 / 140 / 472 / 1063</td>
</tr>
<tr>
<td>RW</td>
<td>801</td>
<td>41 / 676 / 719 / 714</td>
</tr>
</tbody>
</table>
WORD SIMILARITY TASK

• In this task, we compare similarity scores given by models and human annotators

• To measure relationship, Spearman’s rank correlation is considered

• Results compared with human annotators rankings
• HMSN gives much better performance for datasets with frequent words (WS353, MC, RG)

• C&W performs much better with the rare words datasets (SCWS*, RW)

• csmRNN outperforms cimRNN in every case!
RESULT AND EVALUATION (CONTD.)

- Syntactically, cimRNN enforces structural agreement - ‘JJ-ness’ and ‘fearlessness’
- Considering semantics, words that share same stem are clustered together, not good!
- csmRNN model seems to have a good balance between the two features, mainly thanks to using the context information
THANKS FOR ATTENTION!

• Any questions?