CNRS - INPT - UPS - UT1 - UTM

A Neural Network Approach to Selectional Preference Acquisition

Tim Van de Cruys
CNRS & IRIT, Toulouse, France

EMNLP – DOHA, QATAR – 26 October 2014

Introduction

- Predicates often have a semantically motivated preference for particular arguments

(1) The vocalist sings a ballad.

(2) *The exception sings a tomato.

→ known as *selectional preferences*

Introduction

- majority of language utterances occur very infrequently
- models of selectional preference need to properly generalize
- Earlier approaches:
  - hand-crafted resources (WordNet)
  - latent variable models
  - distributional similarity metrics
- this research: *neural network model*

Model overview

- Inspired by recent advances of neural network models for NLP applications [Collobert and Weston 2008]
- Train a neural network to discriminate between felicitous and infelicitous arguments for a particular predicate
- Entirely unsupervised: preferences are learned from corpus data
  - positive instances constructed from attested corpus data
  - negative instances constructed from randomly corrupted data
- two network architectures: for both two-way and multi-way preferences

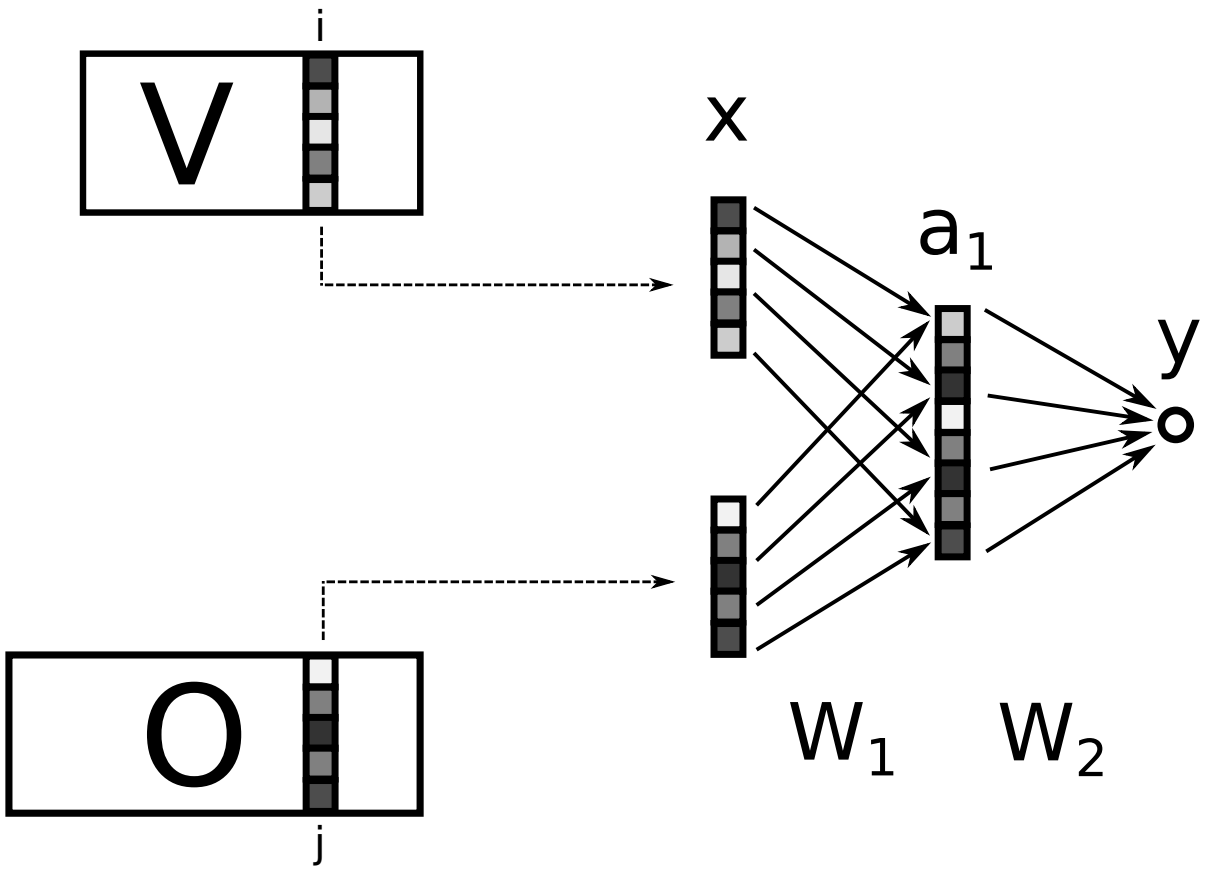
Neural network architecture

- feed-forward neural network architecture with one hidden layer
- tuple (i, j) is represented as concatenation of vectors $\mathbf{v}_i$ and $\mathbf{o}_j$, extracted from embedding matrices $\mathbf{V}$ and $\mathbf{O}$ (learned during training)
- Vector $\mathbf{x}$ then serves as input vector to our neural network.

$$
\begin{aligned}
\mathbf{x} &= [\mathbf{v}_i, \mathbf{o}_j] \\
\mathbf{a}_1 &= f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \\
y &= \mathbf{W}_2 \mathbf{a}_1
\end{aligned}
$$

- $\mathbf{a}_1$: activation of hidden layer
- $\mathbf{W}_1$ and $\mathbf{W}_2$: first and second layer weights
- $\mathbf{b}_1$: first layer's bias
- $f(\cdot)$: element-wise activation function tanh

Graphical representation



Training

- Proper estimation of neural network's parameters requires large amount of training data
- Create unsupervised training data by corrupting actual attested tuples
- Cost function that learns to discriminate between good and bad examples (margin of at least one)

$$\sum_{j' \in J} \max(0, 1 - g[(i,j)] + g[(i,j')])$$

- Compute derivative of the cost with respect to the model's parameters
- Update parameters through backpropagation

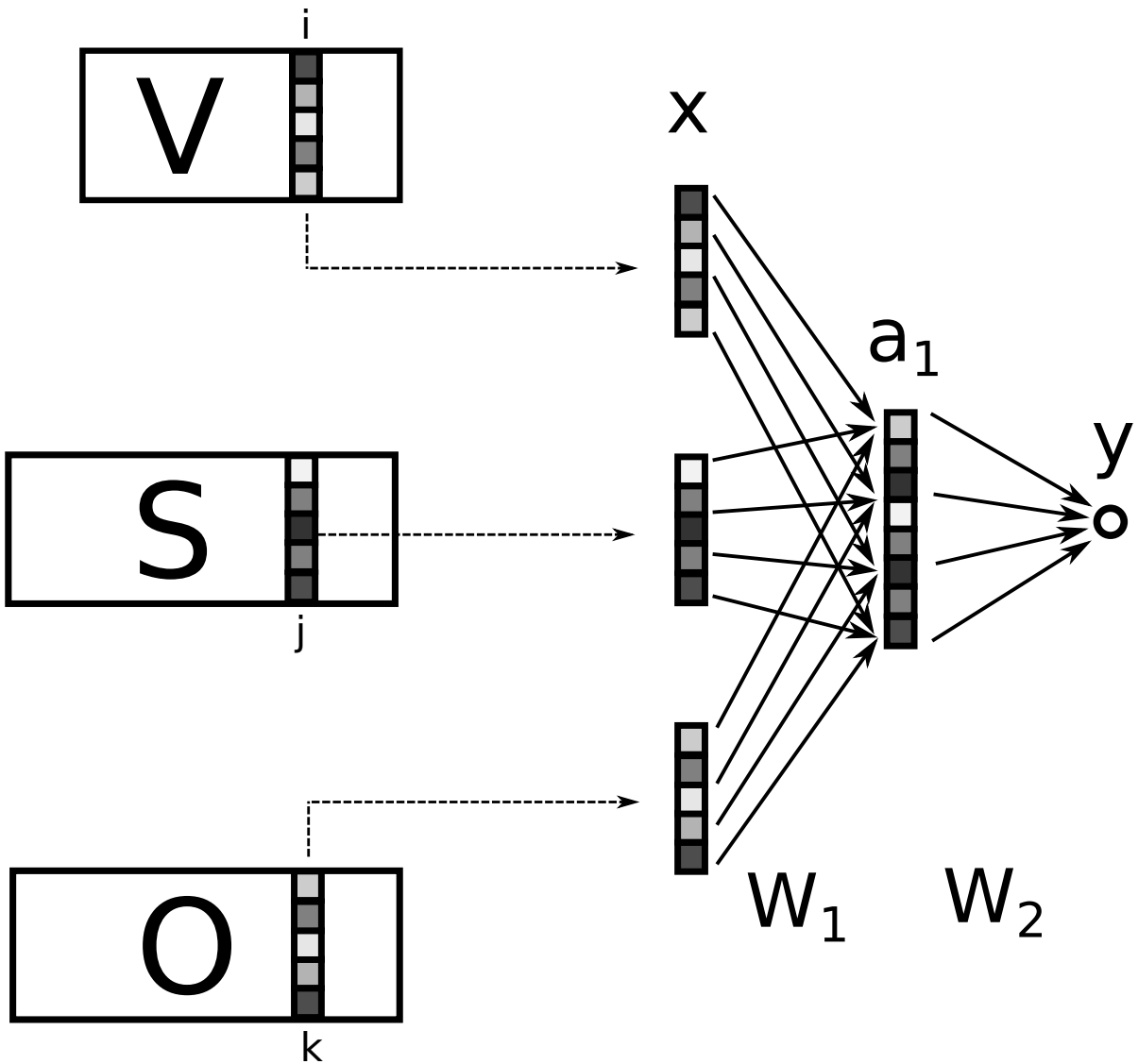
Multi-way selectional preferences

- Similar to two-way case, but one extra embedding matrix for each extra argument
- E.g., for *subject-verb-object* tuples, input vector is

$$\mathbf{x} = (\mathbf{v}_i, \mathbf{s}_j, \mathbf{o}_k)$$

- Rest of the network architecture stays the same

Graphical representation



Training

- Adapted version of training objective
- Given attested tuple (i, j, k), discriminate the correct tuple from corrupted tuples (i, j, k'), (i, j', k), (i, j', k')

$$\sum_{k' \in K} \max(0, 1 - g[(i, j, k)] + g[(i, j, k')])$$
$$+ \sum_{j' \in J} \max(0, 1 - g[(i, j, k)] + g[(i, j', k)])$$
$$+ \sum_{\substack{j' \in J \\ k' \in K}} \max(0, 1 - g[(i, j, k)] + g[(i, j', k')])$$

Implementational details

- Evaluate two-way model using *verb-object* pairs, multi-way model using *subject-verb-object* triples
- applied to English, using UKWAC corpus (2B words, parsed using MaltParser)
- 7K verbs $\times$ 30K objects, 2K verbs $\times$ 10K subjects $\times$ 10K objects
- matrix embedding size of 50; 100 units in hidden layer
- mini-batch L-BFGS with 1000 pairs of good and corrupt tuples per batch for training

Evaluation

- pseudo-disambiguation task to test generalization capacity (standard automatic evaluation for selectional preferences)

v	s	o	s'	o'
win	*team*	*game*	*diversity*	*egg*
publish	*government*	*document*	*grid*	*priest*
develop	*company*	*software*	*breakfast*	*landlord*

- 10-fold cross validation

Results: two-way

model	accuracy ($\mu \pm \sigma$)
[Rooth et al. 1999]	.720 $\pm$ .002
[Erk et al. 2010]	**.887** $\pm$ .004
2-way neural network	.880 $\pm$ .001

- Slightly better result of model based on distributional similarity
- But: Erk et al.'s model is very slow, neural network model is very fast

Results: three-way

model	accuracy ($\mu \pm \sigma$)
[Van de Cruys 2009]	.868 $\pm$ .001
3-way neural network	**.889** $\pm$ .001

- Neural network approach reaches state-of-the-art results for multi-way selectional preference induction

Examples

DRINK	PROGRAM	INTERVIEW	FLOOD
SIP	RECOMPILE	RECRUIT	INUNDATE
BREW	UNDELETE	PERSUADE	RAVAGE
MINCE	CODE	INSTRUCT	SUBMERGE
FRY	IMPORT	PESTER	COLONIZE

Examples

PAPER	RASPBERRY	SECRETARY	DESIGNER
BOOK	COURGETTE	PRESIDENT	PLANNER
JOURNAL	LATTE	MANAGER	PAINTER
ARTICLE	LEMONADE	POLICE	SPECIALIST
CODE	OATMEAL	EDITOR	SPEAKER

Examples

WALL	PARK	LUNCH	THESIS
FLOOR	STUDIO	DINNER	QUESTIONNAIRE
CEILING	VILLAGE	MEAL	DISSERTATION
ROOF	HALL	BUFFET	PERIODICAL
METRE	MUSEUM	BREAKFAST	DISCOURSE

Examples

- Separate word representations for subject and object position
- Allows model to capture specific characteristics for words given their argument position
  - *virus*
    - subject slot: similar to active words like *animal*
    - object slot: similar to passive words like *cell*, *device*
  - *mouse*
    - subject slot: similar to *animal*, *rat*
    - object slot: similar to *web*, *browser*

Conclusion

- neural network architecture that learns to discriminate between felicitous and infelicitous predicate-argument tuples (both two-way and multi-way) in an entirely unsupervised way
- state-of-the-art and very fast once trained

Future work

- Incorporate information from other sources (e.g. initialize embedding matrices with distributed representation from a large-scale neural language model)
- Investigate advantages and disadvantages of having different embedding matrices for different argument positions in multi-way case
- Investigate more advanced neural network architectures, in particular neural tensor networks


Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.


Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.


Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 104–111. Association for Computational Linguistics.


Tim Van de Cruys. 2009. A non-negative tensor factorization model for selectional preference induction. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 83–90, Athens, Greece, March. Association for Computational Linguistics.