Applying Backpropagation Networks to Anaphor Resolution

Roland Stuckardt
Johann Wolfgang Goethe University Frankfurt

Knowledge-poor anaphor resolution ...

- rule-based approaches:
  - Lappin & Leass (1994)
  - Kennedy & Boguraev (1996)
  - Baldwin (1997)
  - Mitkov (1998)
  - …
- corpus-based approaches:
  - Connolly et al. (1994): Naïve Bayes, d. trees, neural networks, ...
  - Aone & Bennett (1995): decision trees
  - Ge et al. (1998): Naïve Bayes
  - Soon et al. (2001): decision trees
  - Ng & Cardie: decision trees (2002), Naïve Bayes (2003)
  - …

… not much research on neural networks

- survey by Olsson (2004):
  only **Connolly et al (1994)** investigate **neural networks**
- Connolly et al (1994): object (NP) anaphor / coreference resolution
  neural networks better than Naïve Bayes and many other models
  on pronouns, they outperform decision trees
- Grüning & Kibrik (2002):
  neural networks successfully applied for **generating** (= modeling the choice of) referential expressions

→ investigating NN-based AR

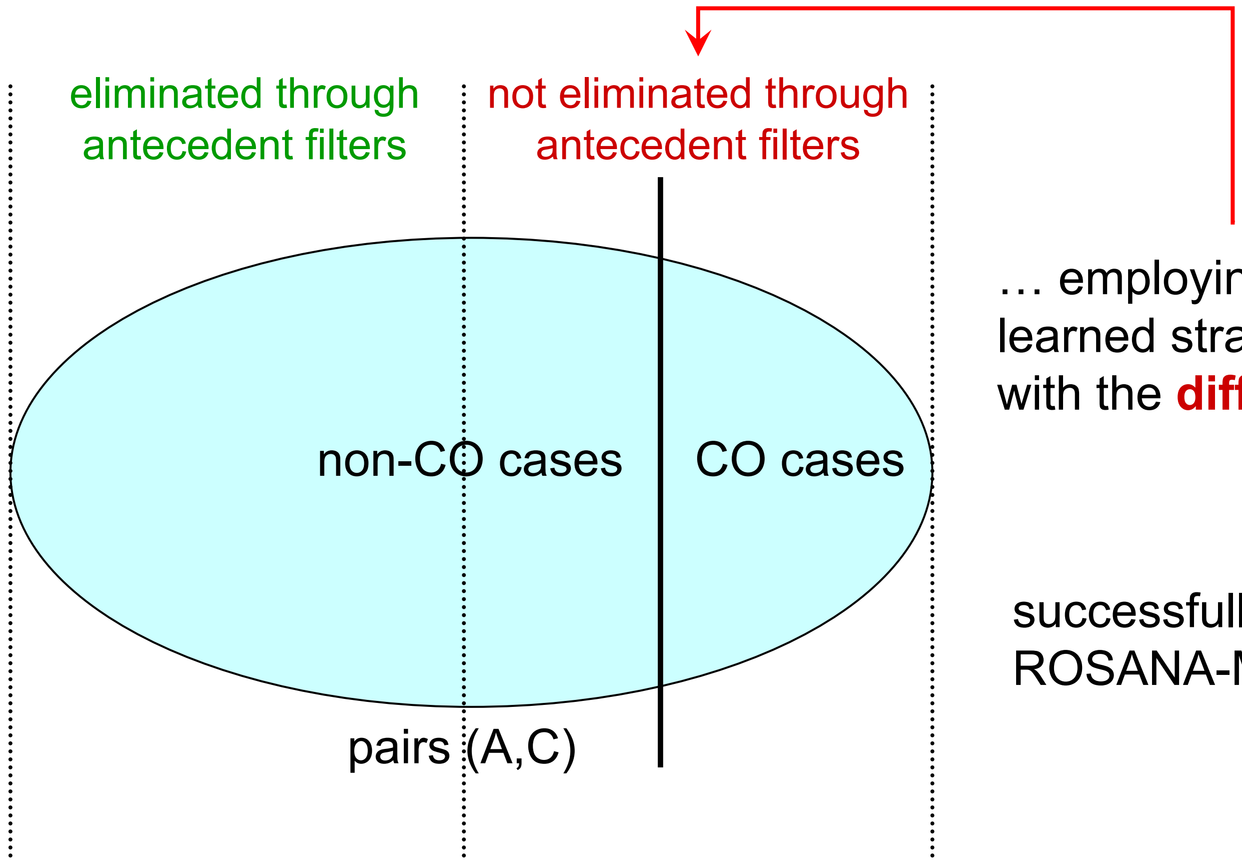
- issues not investigated by Connolly et al (1994):
  - **strategy integration**:
    how to **optimally** make use of machine-learned classifiers for AR
  - **NN configuration optimization**:
    how to **systematically** fine-tune the NN / learning parameters

- points of departure:
  - **ROSANA (2001)**: robust rule-based AR
  - **ROSANA-ML (2002)**: hybrid (partly corpus-based) AR,
    using **decision trees** as antecedent **preference** criteria

→ **ROSANA-NN**

- focusing on third person pronouns

Methodology



… employing machine-learned strategies to deal with the **difficult** cases

successfully applied by ROSANA-ML

Algorithms

Anaphor A, candidates C

- **anaphor resolution**:
  1. apply candidate filters:
     number/gender agreement, syntactic disjoint reference, recency
  2. score and rank remaining candidates according to **NN prediction** and recency
  3. select highest ranking candidate as antecedent

- **training data generation**:
  1. apply candidate filters (according to chosen data generation mode)
  2. generate feature vectors:
     for each remaining candidate C: generate training case fv(A,C)
  3. classify training cases fv(A,C) by consulting annotated corpus (→ fv(A,C)::**K**)

- **neural network learning**:
  1. **learn backpropagation network** over the classified training cases (implementation of Mitchell (2004))

Formal AR evaluation

text corpus for training and evaluation:

- 53 referentially annotated press releases (24,886 tokens)
- 332 third-person non-possessives
  212 third-person possessives
- partitioned into 6 document sets of approximately equal size

no intellectual intervention:

- all experiments on potentially noisy data
- robust preprocessor: FDG parser for English
  (Järvinen & Tapanainen)

two **AR evaluation disciplines**: accuracies

A_{ia}	**i**mmediate **a**ntecedents	*she* ← *her*
A_{na}	**n**on-pronominal **a**nchors	*Merkel* ← *her*

Dealing with the experimental degrees of freedom

parameters:

- features, feature vector signatures
- size of hidden layer
- training data generation settings
  → distribution of positive and negative training cases
- number of training epochs
- I/O encoding
- learning rate and momentum

… to be empirically optimized based on cross-validation:

- extrinsic: AR (antecedent selection) accuracy A_{ia}
- intrinsic: learned classifiers' accuracy (A_{C+N}, A_C)

→ two experimental stages:

- **stage 1:**
  - training data generation modes
  - features, signatures
- **stage 2:**
  - training data generation modes ff
  - size of hidden layer
  - number of training epochs

cross-validation at stage 2 only:

expectation that the first, **coarse** narrowing down of the settings can be performed WLOG on a particular (training, evaluation) set partition

Stage 1: training cases

six **training data generation modes**:
which pairs (A,C) to consider for generating training cases **fv(A,C)::K**

- ***standard***: pairs (A,C) as considered in step 2 of the AR algorithm
- ***no recency filter***
- ***SNL*** (Soon et al., 2001): for each A, at most one positive sample: the nearest cospecifying C_{Co}; all negative cases C_{No} inbetween
- ***NC*** (Ng and Cardie 2002, 2003): as ***SNL***, but C_{Co} non-pronominal
- ***no cataphors***
- ***no cataphors & no recency filter***

Angela Merkel$_{R2}$ …	*President Bush*$_{R1}$ … *Berlin*$_{R3}$ …	*he*$_{R1}$ … *Washington*$_{R4}$ … *she*$_{R2}$ … ***Bush***$_{R1}$
no recency filter	***NC***	***SNL***

Stage 1: sources of evidence

20 robustly computable **features**:

feature	*examples of instances*	*#IN*
type (O)	**PER3, POS3, NAME, CN, …**	**16**
synfun (O)	**subje, trans, …**	**16**
number (O)	**SG, PL, SGPL**	**2**
gender (O)	**MA, FE, NEU, MAFE, …**	**3**
dist (A,C)	**INTRA, PREV, PPREV**	**3**
synpar (A,C)	**YES, NO**	**1**
subject (O)	**YES, NO**	**1**
pronoun (C)	**YES, NO**	**1**
theNP (C)	**YES, NO**	**1**
…	**…**	**…**

A = anaphor,
C = candidate,
O in { A, C }

→ experiments with 6 **signatures**

Stage 1: results

training set: $d_1^{53} – d_{s6}$
evaluation set: d_{s6}

results:

- signature s_e (18 features, 79 inputs):
  - with dgms *SNL, NC:* CO accuracy $A_C > 0.5$
  - with dgm *SNL*: highest A_C of 0.68 on non-possessives

→ **at stage 2:**

- signature $\mathbf{s_e}$
- dgms ***SNL*** and ***NC*** due to their high A_C
- dgm ***no cataphors*** due to its high overall accuracy A_{C+N}

It remains to be seen whether A_C or A_{C+N} is of higher relevance for AR

Stage 2: hidden layer size, training epochs

intrinsically cross-validated optimization of

- number K of internal nodes, K in {20, 30, 40}
- number T* of training epochs, $0 \leq T^* \leq 1000$ ("*" = "averaged")

→ **4 particularly promising settings** for each pronoun type:

PER3					
setting	**dgm**	**K**	**T***	A_{C+N}	A_C
a	***-cataph.***	**40**	**80**	**0.89**	**0.44**
b	***SNL***	**30**	**740**	**0.85**	**0.54**
c	***NC***	**20**	**700**	**0.86**	**0.62**
d	***-cataph***	**40**	**440**	**0.87**	**0.52**

POS3					
setting	**dgm**	**K**	**T***	A_{C+N}	A_C
A	***-cataph***	**40**	**140**	**0.88**	**0.51**
B	***SNL***	**30**	**500**	**0.81**	**0.59**
C	***NC***	**20**	**260**	**0.83**	**0.58**
D	***SNL***	**30**	**40**	**0.86**	**0.45**

Stage 2: anaphor resolution

classifier application, 6-fold **extrinsic cross-validation**:

- criterion: immediate antecedents, accuracy A_{ia}

PER3				
a	b	c	d	(against setting A)
0.64	0.60	0.60	0.62	

POS3				
A	B	C	D	(against setting a)
0.71	0.67	0.69	0.74	

- a and D are settings with high **overall** intrinsic A_{C+N}
- → A_C does **not** seem to be of primary importance

→ ultimate results, comparison

… combining the highest scoring settings a and D:

			im. antecedents: A_{ia}		*non-pr. anchors: A_{na}*	
System	**Setting**	**Corpus**	**PER3**	**POS3**	**PER3**	**POS3**
ROSANA-NN	**(a,D)**	**6-cv(d_1^{53})**	**0.64**	**0.74**	**0.61**	**0.64**
ROSANA-ML	**(1_{nc}^{tc},h)**	**6-cv(d_1^{66})**	**0.66**	**0.75**	**0.62**	**0.68**
	(1_{nc}^{tc},h)	**[d_1^{31},d_{32}^{66}]**	**0.65**	**0.76**	**0.62**	**0.73**
ROSANA	**std.**	**[d_1^{31},d_{32}^{66}]**	**0.71**	**0.76**	**0.68**	**0.66**

ROSANA-NN …

- … vs. ROSANA-ML: virtually on a par
- … vs. ROSANA: worse on non-possessives
- … vs. Connolly et al. (1994): A_{na} of **0.62** vs. 0.52
  → ROSANA-NN might thus be ahead

Achievements and findings

- a hybrid AR system ROSANA-NN using backpropagation networks as preference criteria
- a two-stage optimization methodology
- results:
  - backprogagation networks are among the most successful ML models for AR, thus supporting Connolly et al. (1994)
  - backprogagation networks and C4.5 decision trees seem to perform similarly as alternative plug-ins to the hybrid strategy
  - the hybrid ML / rule-based layout of the algorithm might be interpreted as the key success factor
  - rule-based approaches might still be slightly ahead in certain cases

Further research

- evaluating ROSANA-NN on other corpora / text genres
- investigating enhanced NN types,

  e. g. subspace-trained backpropagation networks
- analyzing how classifiers should be biased in order to matc
  the requirements of the particular AR algorithm: towards
  - A_C
  - A_{C+N}
  - A_X ?

→ refined optimization criterion to be referred to at the intrinsic evaluation stages

Thank you!

Appendix

Stage 1: I/O encodings

input encoding, training and application phase:

- binary features: 1 input node
- features with >2 instances: unary encoding
- potentially ambiguous features: unary encoding
- 0.1 at activated input(s),
  0.9 at the other inputs

output encoding, training phase:

- 0.9, if cospecifying;
- 0.1, if not cospecifying.

output **interpretation**, application phase:

- $>0.5 \rightarrow$ CO (to be **preferred** during antecedent selection)
- $\leq 0.5 \rightarrow$ NON_CO