Applying Backpropagation Networks to Anaphor Resolution

Roland Stuckardt

Johann Wolfgang Goethe University Frankfurt
Im Mellsig 25, D-60433 Frankfurt am Main, Germany
roland@stuckardt.de
http://www.stuckardt.de/

Abstract. Despite some promising early approaches, neural networks have by now received comparatively little attention as a machine learning model for robust, corpus-based anaphor resolution. The work presented in this paper is intended to fill the apparent gap in research. Based on a hybrid algorithm that combines manually knowledge-engineered antecedent filtering rules with machine-learned preference criteria, it is investigated what can be achieved by employing backpropagation networks for the corpus-based acquisition of preference strategies for pronoun resolution. Thorough evaluation will be carried out, thus systematically addressing the numerous experimental degrees of freedom, among which are sources of evidence (features, feature vector signatures), training data generation settings, number of hidden layer nodes, and number of training epochs. According to the evaluation results, the neural network approach performs at least similar to a decision-tree-based ancestor system that employs the same general hybrid strategy.

Key words: anaphor resolution, coreference resolution, machine learning, neural networks, backpropagation networks, decision trees, C4.5, information extraction, discourse, robust natural language processing

1 Introduction

Triggered by pioneering work in the ninetees, the research on robust, operational anaphor resolution has seen a rapid progress in the last decade. Among the knowledge-poor approaches that operate on noisy data are rule-based as well as machine-learning-based systems.[1] A closer analysis reveals that the majority of the corpus-based approaches employs decision trees[2] or Naïve Bayes classifiers[3]. According to the recent survey by Olsson ([13]), there is only the early work of

[1] Among important recent work are the manually designed approaches of Lappin and Leass ([1]), Kennedy and Boguraev ([2]), Baldwin ([3]), Mitkov ([4]), Stuckardt ([5]) and the machine-learning-based approaches Connolly et al. ([6]), Aone and Bennett ([7]), Ge et al. ([8]), Soon et al. ([9]), Stuckardt ([10]).

[2] e. g., Ng and Cardie ([11]), Soon et al. ([9]), Aone and Bennett ([7])

[3] e. g., Ng and Cardie ([12]), Ge et al. ([8])

Connolly et al. ([6]) that investigates neural networks as a device for coreference resolution.

Notably, the research of Connolly et al. ([6]) gave evidence that neural networks, employed as classifiers making coreference predictions for instances of object (NP) anaphora, yield better results than other, less complex ML techniques, among which are Naïve Bayes and Posterior classifiers; regarding pronominal anaphors, formal evaluation indicated that neural networks might even outperform decision trees. In light of these promising early results, the question arises why neural networks have been largely neglected by subsequent research and, in particular, why the majority of approaches to ML-based anaphor and coreference resolution focused on decision trees or Naïve Bayes techniques. Moreover, there have been recent successful applications of neural networks to the problem of modeling the choice of referential expressions (e. g., Grüning and Kibrik, [14]). This too hints towards a closer examination of neural networks for anaphor resolution, since the issues of generation and interpretation can be regarded to be closely related: if, in a certain context, the model of referential choice predicts the usage of a *pronominal* expression for mentioning a particular discourse referent, this might as well be interpreted as evidence for chosing the discourse referent as antecedent for a pronoun occurring in this context.

The work presented below is intended as a first step towards closing this apparent gap in research. While chiefly comparing different machine learning models with respect to the application case of anaphor resolution, Connolly et al. ([6]) neglected a bunch of further important issues, among which are the empirical fine-tuning of the neural network learning parameters, the strategy employed for training data generation, and the sources of evidence to be taken into account, or how to optimally integrate machine-learned classifiers into a fully-fledged anaphor resolution algorithm (see Mitkov, [15]). However, in order to obtain expressive evaluation results that properly compare with the results of other state-of-the-art approaches, these points should be addressed as well.

Two previous studies are taken as the points of departure: [5], describing ROSANA, a classical salience-based and manually knowledge-engineered algorithm for robust pronominal anaphor resolution, and [10], describing the descendant system ROSANA-ML, in which the salience-based preference rankings are substituted by classifiers that are automatically acquired through C4.5 decision tree learning. In the current investigation, a system ROSANA-NN will be designed and evaluated that employs the same general algorithm as ROSANA-ML, but uses neural networks instead of decision trees for antecedent candidate ranking. Thus, as elaborated in [10], the conceptually clean distinction between domain- and genre-independent restrictions and at least partly genre-specific antecedent selection preferences provides an adequate base for the focused application of machine-learned (here: neural network) classifiers as part of a hybrid strategy in which the universal filtering criteria remain manually engineered. While the experiments thus consider the task of robust pronominal anaphor resolution, the general scope of the conducted research is much broader. The fundamental strategy for integrating anaphor interpretation criteria as well as the neural network learning framework developed below apply to the great majority of anaphora types. The paper should thus be conceived as contributing much more than yet another robust pronoun resolver.

The presentation of the work is organized as follows: section 2 provides a general description of the methodology as well as the algorithms and systems used for training data generation, neural network learning, and classifier application. In section 3, the different experimental stages to be carried out are identified; clearly, they are directly related to the plethora of configuration options of the type of classifiers (here: neural networks trained by a backpropagation algorithm) to be learned. Employing this experimental framework, section 4 then presents the evaluation results and the empirical findings. In section 5, the results of ROSANA-NN will be compared with the results of competing approaches, looking at its decision-tree-based and manually knowledge-engineered ancestors ROSANA-ML and ROSANA as well as at the work of Connolly et al. ([6]). Finally, in section 6, conclusions are drawn and directions of further research are identified.

2 Methodology and Algorithms

According to the employed neural-network-based machine-learning approach, two phases are distinguished. (1) During the *training phase*, based on a training text corpus, a set of feature vectors is generated which consists of feature tuples derived from the *(anaphor, antecedent candidate)* pairs that are still considered during the antecedent *selection* phase of the anaphor resolution algorithm, i. e. pairs that have passed all (strict) antecedent filtering criteria. By employing intellectually gathered key data, these vectors are then classified as either cospecifying or non-cospecifying. In the classifier learning step proper, these training cases are submitted to Mitchell's implementation of the backpropagation algorithm [16][4], which, by employing a gradient descent learning strategy and a feedforward technique, iteratively adjusts the weights of a multi-layer neural network with the goal to converge towards a classifier properly fitting the training data and suitable for accurately categorizing unseen feature vectors that are of the same signature as the training vectors. (2) In the *application (anaphor resolution) phase*, the learned classifiers are employed for antecedent selection: to discern between more and less plausible candidates, instead of salience factors, neural network classifiers are applied. Thus, as initially motivated, the basic strategy consists in learning the preference criteria only, thus resorting to classical rule-based robust implementations of the antecedent filtering strategies, among which are syntactic disjoint reference and number/gender agreement.

Two algorithms are hence distinguished: (a) the feature vector generation algorithm, which is employed during the training phase, and (b) the anaphor resolution algorithm proper, which specifies the general strategy of the application phase.

[4] See chapter 4 of [16]; the backpropagation implementation has been taken from the webpage `http://www.cs.cmu.edu/~tom/mlbook.html` (December 2004).

2.1 Feature Vector Generation

In figure 1, the specification of the feature vector generation algorithm is given. Step 1, in which different kinds of restrictions for eliminating impossible antecedents (in particular, agreement in person/number/gender and syntactic disjoint reference) are applied, is identical with the antecedent filtering phase of the manually designed ROSANA algorithm. In step 2, however, during feature vector generation, the salience ranking of the antecedent candidates is substituted by the mapping of each remaining anaphor-candidate pair (α_i, γ_j) to a feature vector $fv(\alpha_i, \gamma_j)$, the attributes $f_1, \ldots, f_{k_\sigma}$ of which comprise individual and relational features derived from the descriptions of the occurrences α_i and γ_j. The *signature* of the feature vectors, i. e. the inventory of features to be taken into account, has to be chosen carefully in order to fulfill the conditions of robust processing: instead of requiring complete and unambiguous descriptions, they should be computable from potentially partial representations such as fragmentary syntactic parses. (See section 4.1.)

1. *Candidate Filtering*: for each anaphoric NP α, determine the set of admissible antecedents γ:
   (a) verify morphosyntactic (number and gender) or lexical agreement with γ;
   (b) if the antecedent candidate γ is intrasentential: apply the robust syntactic disjoint reference filter as specified in [5], figure 4.
2. *Feature vector generation*: for each remaining anaphor-candidate pair (α_i, γ_j):
   (a) generate, according to the feature signature σ under consideration, the feature vector
   $$fv(\alpha_i, \gamma_j) := (n_{\alpha_i}, n_{\gamma_j}, f_1, \ldots, f_{k_\sigma}).$$
   where n_{α_i} and n_{γ_j} are the numbers (unique identifiers, referred to in the key) of the occurrences α_i and γ_j, and $f_1, \ldots, f_{k_\sigma}$ are (individual and relational) features derived from α_i and γ_j with respect to the signature σ;
   (b) write $fv(\alpha_i, \gamma_j)$ to an external training data file.

Fig. 1. ROSANA-NN: feature vector generation

2.2 Anaphor Resolution

The specification of the ROSANA-NN anaphor resolution algorithm proper is given in figure 2. Again, step 1 is identical with the antecedent filtering phase of the manually designed ROSANA algorithm. Step 2, however, is modified. For a particular instance (α_i, γ_j) of anaphor and antecedent candidate, after the computation of the feature vector $fv(\alpha_i, \gamma_j)$, a learned neural network classifier, which might depend upon the the particular type of anaphor to be resolved, is consulted;[5] basically, its result $\Psi_\sigma^{type(\alpha_i)}(fv(\alpha_i, \gamma_j))$ consists in a prediction

[5] By now, due to technical reasons, the classifier application has not been technically integrated with the ROSANA-NN implementation; rather, the consultation is ac-

1. *Candidate Filtering*: for each anaphoric NP α, determine the set of admissible antecedents γ:
   (a) verify morphosyntactic (number and gender) or lexical agreement with γ;
   (b) if the antecedent candidate γ is intrasentential: apply the robust syntactic disjoint reference filter as specified in [5], figure 4.
2. *Candidate scoring and sorting*:
   (a) for each remaining anaphor-candidate pair (α_i, γ_j):
      i. *consultation of the neural network classifier*: determine the prediction $\Psi_\sigma^{type(\alpha_i)}(fv(\alpha_i, \gamma_j))$ of the learned neural network classifier with respect to the instance $fv(\alpha_i, \gamma_j)$.
   (b) for each anaphor α: sort candidates γ_j according the following criteria:
      - *primary*: candidates γ_j for which $\Psi_\sigma^{type(\alpha)}(fv(\alpha, \gamma_j)) = \mathit{COSPEC}$ are preferred over candidates $\gamma_{j'}$ for which $\Psi_\sigma^{type(\alpha)}(fv(\alpha, \gamma_{j'})) = \mathit{NON\_COSPEC}$;
      - *secondary*: surface nearness.
   (c) sort the anaphors α according to the above criteria applied to their respective best antecedent candidates.
3. *Antecedent Selection*: consider anaphors α in the order determined in step 2c. Suggest antecedent candidates $\gamma_j(\alpha)$ in the order determined in step 2b.
   Select $\gamma_j(\alpha)$ as candidate if there is no interdependency, i. e. if
   (a) the morphosyntactic features of α and $\gamma_j(\alpha)$ are still compatible,
   (b) for all occurrences $\delta_{\gamma_j(\alpha)}$ and δ_α the coindexing of which with $\gamma_j(\alpha)$ and (respectively) α has been determined in the *current* invocation of the algorithm: the coindexing of $\delta_{\gamma_j(\alpha)}$ and δ_α, which results transitively when choosing $\gamma_j(\alpha)$ as antecedent for α, does neither violate the binding principles nor the i-within-i condition. (see the full specification in [5], figure 4)

Fig. 2. ROSANA-NN: anaphor resolution through backpropagation networks

$\in \{\mathit{COSPEC}, \mathit{NON\_COSPEC}\}$.[6] In the subsequent step, these predictions are employed for ranking the candidate sets of each anaphor: candidates which are (heuristically) classified to cospecify with the anaphor rank higher than candidates that are (heuristically) predicted as non-cospecifying; surface nearness (i. e. word distance) serves as the secondary criterion.[7] There is a final step 3 in which antecedents are selected. The remaining candidates are considered in the order determined by the ranking step; further means are taken to avoid combinations of antecedent decisions that are mutually incompatible (see [5]).

complished by looking up externally precomputed classification results. However, the implementation yields outcomes equivalent to those of a fully integrated system.

[6] To put it formally: a *classifier function* $\Psi_\sigma^{type(\alpha_i)} : A_1 \times A_2 \times \ldots \times A_{k_\sigma} \mapsto \{\mathit{COSPEC}, \mathit{NON\_COSPEC}\}$ is applied that maps instances of the underlying signature σ to cospecification/non-cospecification predictions.

[7] Among the possible refinements are: further ranking the candidates according to the real value ε yielded by the neural network classification result lookup (see section 4.1), or eliminating candidates which are (heuristically) classified as not cospecifying.

3 Layout of Experiments

3.1 Experimental Degrees of Freedom

There are various experimental degrees of freedom that should be considered:

1. the *sources of evidence (features, feature vector signatures)* upon which to classify a given pair (α_i, γ_j) of anaphor and antecedent candidate;
2. the *techniques employed for encoding the input and output space* of the network;
3. the *number κ of internal notes* making up the (here) single hidden neural network layer; this number should be chosen large enough in order to enable the network to learn all relevant regularities of the data space to be modeled; on the other hand, it should not be chosen too large as this might result in an unwanted overfitting of the particular sample data;[8]
4. the *parameters learning rate (η) and momentum (ζ) of Mitchell's backpropagation algorithm* (see [16], p97ff): setting them to low values will drastically slow down network convergence, while choosing them too large might result in missing the sought-for empirical optimum;
5. the *number τ of training epochs*, which should be chosen suitably in order to achieve convergence towards the training data without overfitting them;
6. the settings that determine the *distribution of the training data (data generation mode)*, i. e. the way how positive and negative sample cases are generated based on referentially annotated corpora; this should be addressed by taking into account
7. *the particular way how the classifiers are employed by the anaphor resolution algorithm*, as this determines the distribution of cases relevant during extrinsic classifier application;
8. whether *one general or several anaphor-type-specific classifiers* should be learned.

Thus, there are considerably more dimensions along which one might vary the experimental settings than in case of decision trees (see [10]).

Moreover, in order to obtain results independent of a particular partition of the annotated data into training, validation, and test cases, cross-validation should be carried out:

[8] In general terms, the descriptional capabilities of the representational model and the number of training data should be kept in relation. Regarding neural networks, besides the size of the hidden layer, the chosen feature vector signature as well as the employed encoding strategy determine the number of nodes and, thus, the potential descriptional power of the network. Clearly, the larger the network, the more training data should be available in order to ensure convergence towards a classifier that appropriately generalizes. Notably, this issue of *data fragmentation* is frequently neglected; Ng and Cardie ([11]) mention it briefly with respect to decision tree classifiers, for which they identify the desideratum that each leaf of the learned decision tree should cover roughly the same minimum number of training data instances.

- at *intrinsic (learned classifier) level*, determining the classifiers' accuracy regarding their predictions $\in \{COSPEC, NON\_COSPEC\}$;
- at *extrinsic (application) level*, determining the anaphor resolution results obtained with the classifiers.

3.2 Annotated Text Corpus and Disciplines of Formal Evaluation

The training and evaluation of the ROSANA-NN system will be carried out on a corpus of 53 referentially annotated news agency press releases, comprising 24,886 tokens, 332 third-person non-possessives, and 212 third-person possessive pronouns. In order to support cross-validation, this corpus d_1^{53} has been randomly partitioned into six document sets $ds_i, 1 \leq i \leq 6$ of approximately equal size. In all experiments, the training data generation and the application of the trained system take place on potentially noisy data, i. e. without a-priori intellectual correction of orthographic or syntactic errors, and without any post-editing of the possibly partial or incorrect parses derived by the robust syntactic preprocessor, which is the FDG parser for English of Järvinen and Tapanainen ([17]).

The anaphor resolution performance will be evaluated with respect to two evaluation disciplines. In the *immediate antecedent* (*ia*) discipline, the classical accuracy measure is employed that determines the precision of correct immediate antecedent choices; by further taking into account cases of unresolved pronouns, the respective recall measure is obtained.[9] In the *non-pronominal anchors* (*na*) discipline, antecedents are required to be common or proper nouns, which is particularly relevant for anaphor resolution applications; again, it is distinguished between precision and recall. Thus, the anaphor resolution performance is measured according to the tradeoffs (P_{ia}, R_{ia}) and (P_{na}, R_{na}). For formal definitions and an in-depth discussion of the two measures, the reader is referred to [5].

4 Experiments and Empirical Results

In order to deal with the issues identified in section 3.1, the experiments will be divided into two stages: stage 1, addressing signature optimization, network i/o encoding, and a first, coarse narrowing-down of the data generation settings; stage 2, addressing the issues of identifiying the most promising combinations of data generation mode and number of hidden layer nodes, determining the respective empirically optimal numbers of training epochs, and intrinsically as well as extrinsically evaluating the learned classifiers' performance. In order to limit evaluation efforts, cross-validation will be confined to stage 2; regarding signature and i/o encoding, relative empirical performance is expected to be virtually independent from the particular (training, evaluation) data partitioning.

[9] Under the assumption that *all* pronouns are resolved, the precision measure is equivalent to the accuracy measure employed for evaluating classical approaches of, e.g., Lappin and Leass ([1]) and Kennedy and Boguraev ([2]). By allowing for unresolved pronouns, a (P, R) tradeoff is obtained, which corresponds to the evaluation measure employed by Aone and Bennett ([7]).

Two of the experimental degrees of freedom identified in section 3.1 will not be considered in detail. The question whether to use one general or two type-specific classifiers for non-possessive vs. possessive pronouns has been settled in favour of the latter option, taking into account that the ROSANA-ML experiments have brought evidence that type-specific classifiers might yield slighty better results if combined with appropriate training data generation strategies (see [10]). Likewise, first experiments have indicated that the backpropagation parameters of learning rate (η) and momentum (ζ) should be kept best at their original settings ($\eta = 0.3$ and $\zeta = 0.3$, see [16], p97ff).

4.1 Stage 1: Data Generation, Signatures, and I/O Encodings

At the first stage of experiments, the number κ of internal (hidden layer) nodes is set to the fixed value of 20. Each training run is confined to $\tau = 160$ training epochs. The goal consists in determining a promising subset of signatures and data generation modes to be evaluated in full detail at the second experimental stage, where the parameters κ and τ will then be reconsidered.

Data Generation Modes Six data generation modes are considered, four of which have already been investigated in the ROSANA-ML decision tree experiments. The data generation modes differ with respect to the subset of (α, γ) occurrence pairs used for generating positive (classified as COSPEC) and negative (classified as NON_COSPEC) training cases:

- *standard*: the set of antecedent candidates γ to be paired with a particular anaphor α for generating training vectors $fv(\alpha, \gamma)$ is identical with the set of candidates considered by the ROSANA-NN anaphor resolution algorithm in its canonical configuration; recency filters, which depend upon the type of anaphor to be resolved, apply.
- *no cataphors*: in this case, the same recency filters as under the *standard* setting apply; however, instances of backward anaphora (γ surface-topologically *following* α) are not considered as training samples.[10]
- *no recency filter*: recency filters of the standard setting are switched off; thus, since *all* candidates preceding the anaphor and fulfilling the further filters give rise to a training case, the resulting training set is significantly enlarged; while, from a learning-theoretical point of view, it is considered adequate to mirror the application case distribution as close as possible, this strategy might nevertheless prove reasonable in the (common) case of training data sparsity.
- *no cataphors, no recency filter*: combines the *no cataphors* and *no recency filter* settings.

[10] In fact, the version of the ROSANA-NN algorithm put under scrutiny below also employs a *no cataphors* setting, which might thus be considered as the standard setting proper. However, in order to facilitate comparison, the terminology has been kept identical to the terminology employed in the publications describing the ROSANA-ML results.

- *SNL*: a training data generation strategy successfully applied by Soon et al. ([9]); for each anaphor α, at most one positive sample is included in the training set, viz., the feature vector constructed over α and (as far as existent) its *surface-topologically nearest* cospecifying antecedent γ^N; negative samples are constructed by taking into account all (non-cospecifying) occurrences surface-topologically situated between γ^N and α.
- *NC*: a strategy successfully applied by Ng and Cardie ([11], [12]); as in mode *SNL*, the lookback is restricted by a particular cospecifying antecedent γ^{NP}, which, however, this time is required to be *non-pronominal*; any occurrence between α and γ^{NP} gives rise to a further (here: negative or positive) sample; thus, the data sets derived by applying mode *NC* subsume the respective data sets constructed under mode *SNL*.[11]

Features and Signatures The most fundamental question regards the set of attributes, i. e. the signature of the feature vectors from which the classifiers will be learned. The choice is confined to sources of evidence available in the considered environment of robust, knowledge-poor processing. Figure 3 displays the respective inventory of attributes taken into account during the following experiments. *type(o)* denotes the type of the respective occurrence o, in particular

Feature	Examples of Instances	Description	σ_{DT}	σ_b	σ_c	σ_d	σ_e	#IN
type(α)	PER3, POS3	type of anaphor α	•	•	•			16
synfun(α)	subje, trans	syntactic function of α	•	•		•	•	16
synlevel(α)	TOP, SUB	syntactic position of α	•	•	•	•	•	3
number(α)	SG, PL, SGPL	number of α	•	•	•		•	2
gender(α)	MA, FE, MAFE	gender of α	•	•	•		•	3
type(γ)	NAME, PER3	type of candidate γ	•	•		•	•	16
synfun(γ)	subje, trans	syntactic function of γ	•	•		•	•	16
synlevel(γ)	TOP, SUB	syntactic position of γ	•	•	•	•	•	3
number(γ)	SG, PL, SGPL	number of γ	•	•	•		•	2
gender(γ)	MA, FE, MAFE	gender of γ	•	•	•		•	3
dist(α,γ)	INTRA, PREV	sentence distance	•	•	•	•	•	3
dir(α,γ)	ANA, KATA	resumption direction	•	•	•			1
synpar(α,γ)	YES, NO	syntactic role identity?	•	•		•	•	1
syndom(α,γ)	[$\alpha \rightarrow \gamma$], [$\gamma \rightarrow \alpha$], no	synt. dominance relation	•	•	•	•	•	3
subject(α)	YES, NO	anaphor α is subject?		•	•	•	•	1
subject(γ)	YES, NO	candidate γ is subject?		•	•	•	•	1
pronoun(γ)	YES, NO	candidate γ is pronoun?		•	•	•	•	1
thenp(γ)	YES, NO	candidate γ is definite NP?		•	•	•	•	1
prostr(α,γ)	YES, NO	α, γ string-id. pronouns?		•	•	•	•	1
synpar*(α,γ)	SuSP, ObSP, NoSP	weak syntactic role identity		•	•	•	•	3
Σ (#IN)			88	96	47	69	79	96

Fig. 3. inventory of features over which the signatures are defined

[11] Originally, this mode was employed for common NP anaphor resolution.

PER3/POS3 (third person non-possessive/possessive pronouns), VNOM (common noun phrases), and NAME (proper nouns); regarding the anaphor ($o = \alpha$), the choice is restricted to PER3 and POS3 in the current experiments. The feature *synfun(o)* describes the syntactic function of o. *synlevel(o)* captures a coarse notion of (non-relational) syntactic prominence[12], which is measured by counting the number of principal categories[13] occurring on the path between o and the root of the respective parse fragment. Features *number(o)* and *gender(o)* capture the respective morphological and lexical characteristics of anaphor α and candidate γ. Furthermore, some relational features are considered: *dist(α,γ)* (sentence distance, distinguishing between three cases: same sentence, previous sentence, two or more sentences away), *dir(α,γ)* (whether γ topologically precedes α or vice versa), *synpar(α,γ)* (identity of syntactic function)[14], and *syndom(α,γ)* (relative syntactic position of the clauses of anaphor and candidate in case of intrasentential anaphora)[15]. These 14 features constitute the signature σ_{DT} that was found to perform best in the ROSANA-ML decision tree experiments (see [10]).

Motivated by the approaches of Ng and Cardie ([11]) and Soon et al. ([9]), this original inventory of ROSANA-ML attributes has been supplemented by six additional promising features dealing with *pronominal* anaphora and deemed relevant for the acquisition of *preference* strateges.[16] The choice is restricted to those attributes that are computable based on the knowledge made available by the robust preprocessors currently used in the ROSANA-NN framework. This excludes from consideration a bunch of features that deal with semantic class information provided by WordNet.[17] Moreover, in the context of pronominal anaphor resolution, some other attributes, such as BOTH_PRONOUNS by [9] or STR_MATCH by [11] boil down to more trivial features, now dealing merely with the antecedent or with pronoun string identity. Proceeding along these lines, the following six features have been added, five of which are binary: *subject(o)*, capturing whether anaphor/candidate occurs in the subject role; *pronoun(γ)* / *thenp(γ)*, describing whether the candidate is a pronoun or, respectively, a definite NP; the relational feature *prostr(α,γ)*, capturing whether candidate as well as anaphor are pronouns with identical surface form; finally, the ternary relational feature *synpar*(α,γ)* has been introduced, which models a weak version

[12] in contrast to *relational* notions of syntactic prominence, in which the relative position to the other occurrence is taken into account (e.g. *c-command*)

[13] to put it more formally: nodes that, in the sense of the Government and Binding (GB) theory, constitute *binding categories* (see [18])

[14] thus immediately capturing the role inertia information that has been found to be useful in the classical, manually designed approaches [1, 5]

[15] e.g. $[\alpha \rightarrow \gamma]$ describes the case in which the clause of γ is syntactically subordinated to the clause of α

[16] Ng and Cardie ([11]) and Soon et al. ([9]) have a more general scope as they are dealing with common and proper noun anaphora as well, and they are aiming at learning general coreference resolution strategies, including antecedent *filtering* criteria.

[17] Moreover, in the approaches of Ng and Cardie ([11]) and Soon et al. ([9]), this source of evidence primarily addresses cases in which the available semantic information is non-trivial, i. e. cases in which both anaphor and candidate are *non-pronominal.*

of syntactic parallelism, distinguishing three cases: both anaphor and candidate are subjects; neither of them is a subject; exactly one of them is a subject.

I/O Encodings According to figure 3, all considered features take values from a particular finite set. Attributes taking only two values, such as *dir(*α*,*γ*)* or *synpar(*α*,*γ*)*, are binarily encoded by a single network input, assigning an input of 0.1 (TARGET_LOW) in one case and 0.9 (TARGET_HIGH) in the other; attributes defined over a set of more than two values are encoded unarily, thus, for instance, resulting in 16 inputs modeling the *synfun(o)* features as 16 syntactic roles are distinguished, activating exactly one input for a given case and deactivating all other inputs; in the special case of possibly ambiguous attributes such as *number(o)* and *gender(o)*, a canonical powerset encoding scheme is employed. In the rightmost column of figure 3, the number of inputs resulting for the different attributes under this encoding scheme are shown.

In the case of anaphor or coreference resolution, the output encoding happens to be a trivial matter, as the prediction space of the backpropagation network consists only of two elements: *cospecifying* and *non-cospecifying.* During training, the value 0.9 is used to encode the former case, while the value 0.1 is used to model the latter case. During network application, output values > 0.5 are thus interpreted as COSPEC predictions, while values ≤ 0.5 are considered to predict NON_COSPEC cases.

Evaluation Results Distinguishing between classifiers for third person non-possessive and possessive pronouns, intrinsic classifier accuracies have been determined for each combination of the above-defined six data generation modes and five feature signatures; As no cross-validation shall be carried out at experimental stage 1, considerations are restricted to the particular (training, test) data partition $[d_1^{53} \setminus ds_6, ds_6]$.

It would be beyond the scope of the paper to discuss the results for the $2 \cdot 5 \cdot 6 = 60$ combinations in full detail. However, there are some important findings that shall be briefly summarized as they give rise to focus on one particularly promising signature and three auspicious data generation modes at experimental stage 2. Considerations shall be restricted to signature σ_e and modes *SNL*, *NC*, and *-ca (no cataphors)*, as it turned out that: (1) with only one exception, modes *SNL* and *NC* yield the relatively highest C accuracies; (2) (only) for signatures σ_{DT}, σ_b, and σ_e, both *SNL* and *NC* achieve an outstanding C accuracy well above the 50% level; thus, if one suspects that high C accuracy might be relevant for improving extrinsic (anaphor resolution) performance, these three signatures and two modes are on the short list; (3) regarding the important case of PER3 pronouns, the combination of σ_e and *SNL* achieves a particularly high C accuracy of 0.68; thus, it has been decided to focus on signature σ_e in the subsequent experiments;[18] finally, in order to cover classifiers biased towards C $\cup$ N accuracy, mode *-ca* shall be considered as well as it exhibits high C $\cup$ N accuracy while still yielding a reasonable C accuracy.

[18] Using different signatures for PER3 and POS3 classifiers might be a further option.

4.2 Stage 2: Internal Nodes, Training Epochs, and Cross-Validation

With the goal of systematically narrowing down the remaining set of experimental options, stage 2 deals with: (1) optimizing the number τ of training epochs, which was provisionally limited to 160 at stage 1; (2) selecting an appropriate number κ of internal (hidden layer) nodes, which was provisionally set to the fixed value of 20 at stage 1. In this context, the requirement *systematically* amounts to empirically optimizing the parameters τ and κ based on intrinsic cross-validation runs. Eventually, a small number of particularly promising configurations shall be identified and subjected to the ultimate discipline of (3) extrinsic (ROSANA-NN anaphor resolution) cross-validation.

Intrinsic Cross-Validation: Training Epochs (τ), Internal Nodes (κ) The issue of avoiding overfitting the training data shall be addressed by an intrinsic cross-validation approach according to which τ is set to the average $\tau^* = \frac{1}{6}\sum_{i=1}^{6}\tau_i$ over the six data partitions. Thus, there are actually two substages of intrinsic cross-validation: the first one employed to determine average values τ^*; the second one carried out to determine the intrinsic classifier performance.

Separate experiments shall be carried out for each combination of data generation mode and considered number κ of internal network nodes. Hidden layers of three sizes will be considered: $\kappa \in \{20, 30, 40\}$. As it is further distinguished between non-possessives vs. possessives and C ∪ N vs. C accuracy, $3 \cdot 3 \cdot 2 \cdot 2 = 36$ intrinsic cv experiments are carried out at both substages.

Figure 4 displays the results of the first substage of intrinsic cross-validation, viz., the average numbers τ^* of epochs to train before a worsening of the respective ((C ∪ N) or C) accuracy on the test data indicates that the learned network begins to overfit the training data. As the above experiments indicated that 160 training cycles might in general be insufficient, the considered interval of epochs is enlarged to $0 \leq \tau \leq 1000$. These results confirm the tendency observed at the above experimental stage 1: without exception, it takes more training cycles to converge towards a classifier with high C accuracy (columns τ^*_C) than towards a classifier with high C ∪ N accuracy (columns $\tau^*_{C\cup N}$), and with only one exception (POS3, *-ca*, $\kappa = 20$), the difference regarding the appropriate number of training cycles is considerable. Moreover, the above preliminary upper bound of $\tau \leq 160$ turns out to be too small to learn empirically optimal classifiers biased towards high C accuracy.

PER3	$\kappa = 20$		$\kappa = 30$		$\kappa = 40$	
	$\tau^*_{C\cup N}$	τ^*_C	$\tau^*_{C\cup N}$	τ^*_C	$\tau^*_{C\cup N}$	τ^*_C
-ca	260	480	100	340	80	440
SNL	100	560	80	740	100	680
NC	60	700	60	500	60	300

POS3	$\kappa = 20$		$\kappa = 30$		$\kappa = 40$	
	$\tau^*_{C\cup N}$	τ^*_C	$\tau^*_{C\cup N}$	τ^*_C	$\tau^*_{C\cup N}$	τ^*_C
-ca	300	340	200	460	140	500
SNL	40	700	40	500	40	540
NC	160	260	40	240	20	280

Fig. 4. PER3 and POS3 classifiers: average values τ^* (cross-validated)

Turning towards substage 2, viz., intrinsic cross-validation proper, the figures obtained for the overall 36 experiments generally confirm the results obtained at stage 1 on the particular (training, test) data partition $[d_1^{53} \setminus ds_6, ds_6]$. For each pronoun type, intrinsic cross-validation results of four particularly promising *(dgm, κ)* combinations are displayed in figure 5;[19] these are the combinations that will be further subjected to extrinsic cross-validation below. Regarding non-possessives, combination *a* is considered because of its outstanding C ∪ N accuracy, while *b* and *c* are chosen because of their high C accuracy; combination *d* is included because it promises a relatively high C accuracy which is expected to be accompanied by relatively high C ∪ N results. Concerning the possessive pronouns, combinations *B* and *C* are considered because of their outstanding C accuracy, while *A* and *D* are selected because they promise high C ∪ N results which are expected to come with a still relatively high C performance.

Setting		DGM	κ	τ^*	$A_{C \cup N}$	A_C
PER3	*a*	*-ca*	40	80	**0.89**	0.44
	b	*SNL*	30	740	0.85	**0.54**
	c	*NC*	20	700	0.86	**0.62**
	d	*-ca*	40	440	0.87	**0.52**

Setting		DGM	κ	τ^*	$A_{C \cup N}$	A_C
POS3	*A*	*-ca*	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

Fig. 5. PER3 and POS3 classifiers: results of intrinsic cross-validation

According to the results, some of the combinations seem to be unattractive as they are quantitatively outperformed by one of their competitors; this holds with respect to the PER3 *b* setting, which is majorized by PER3 *c*, and with respect to POS3 *D*, which is majorized by POS3 *A*. However, beyond the merely quantitative aspects, there are the qualitative issues of data distribution: classifiers should perform well on the particular subset of cases most relevant for anaphor resolution. Hence, these settings shall be further considered anyway, as they might differ substantially regarding the distribution of the correctly classified cases. In fact, one criterion governing the selection of the above combinations has been to consider each data generation mode (*-ca*, *SNL*, and *NC*) and each optimization criterion (A_C and $A_{C \cup N}$) at least once for both pronoun types.

Extrinsic Cross-Validation In figures 6 and 7, the results of the 6-fold extrinisic (anaphor resolution) cross-validation experiments are displayed. The tables show the results for each particular setting in the discipline of immediate antecedency (*ia*), given as tradeoffs (P_{ia}, R_{ia}).[20] Since, in all but one case,

[19] Due to space limitations, results on the particular data partitions are not included.

[20] As ROSANA-NN intertwines PER3 and POS3 resolution, there might be interdependencies between these two subprocesses, which implies that it is impossible to extrinsically evaluate them separated from each other. In fact, the results given for the non-possessives have been obtained by employing the classifiers pertaining to settings *a*, *b*, *c*, *d* together with the setting *A* classifier as the *standard classifier for*

	(P_{ia}, R_{ia})			
	a	b	c	d
(ds1) $[d_1^{53} \setminus ds_1, ds_1]$	$(0.58, 0.58)$	$(0.58, 0.58)$	$(0.53, 0.53)$	$(0.63, 0.63)$
(ds2) $[d_1^{53} \setminus ds_2, ds_2]$	$(0.64, 0.64)$	$(0.59, 0.59)$	$(0.59, 0.59)$	$(0.69, 0.69)$
(ds3) $[d_1^{53} \setminus ds_3, ds_3]$	$(0.67, 0.67)$	$(0.58, 0.58)$	$(0.63, 0.63)$	$(0.60, 0.60)$
(ds4) $[d_1^{53} \setminus ds_4, ds_4]$	$(0.71, 0.70)$	$(0.67, 0.66)$	$(0.70, 0.69)$	$(0.63, 0.63)$
(ds5) $[d_1^{53} \setminus ds_5, ds_5]$	$(0.59, 0.59)$	$(0.59, 0.59)$	$(0.55, 0.55)$	$(0.59, 0.59)$
(ds6) $[d_1^{53} \setminus ds_6, ds_6]$	$(0.63, 0.63)$	$(0.61, 0.61)$	$(0.61, 0.61)$	$(0.57, 0.57)$
(ds1-6): weighted avg.	$(\mathbf{0.64}, \mathbf{0.64})$	$(0.60, 0.60)$	$(0.60, 0.60)$	$(0.62, 0.61)$

Fig. 6. PER3 classifiers, 6-fold extrinsic (anaphor resolution) cv (*ia* discipline)

	(P_{ia}, R_{ia})			
	A	B	C	D
(ds1) $[d_1^{53} \setminus ds_1, ds_1]$	$(0.70, 0.70)$	$(0.58, 0.58)$	$(0.55, 0.55)$	$(0.76, 0.76)$
(ds2) $[d_1^{53} \setminus ds_2, ds_2]$	$(0.67, 0.67)$	$(0.67, 0.67)$	$(0.67, 0.67)$	$(0.75, 0.75)$
(ds3) $[d_1^{53} \setminus ds_3, ds_3]$	$(0.76, 0.76)$	$(0.85, 0.85)$	$(0.73, 0.73)$	$(0.79, 0.79)$
(ds4) $[d_1^{53} \setminus ds_4, ds_4]$	$(0.75, 0.75)$	$(0.64, 0.64)$	$(0.77, 0.77)$	$(0.74, 0.74)$
(ds5) $[d_1^{53} \setminus ds_5, ds_5]$	$(0.70, 0.70)$	$(0.63, 0.63)$	$(0.74, 0.74)$	$(0.74, 0.74)$
(ds6) $[d_1^{53} \setminus ds_6, ds_6]$	$(0.66, 0.66)$	$(0.66, 0.66)$	$(0.63, 0.63)$	$(0.69, 0.69)$
(ds1-6): weighted avg.	$(0.71, 0.71)$	$(0.67, 0.67)$	$(0.69, 0.69)$	$(\mathbf{0.74}, \mathbf{0.74})$

Fig. 7. POS3 classifiers, 6-fold extrinsic (anaphor resolution) cv (*ia* discipline)

every PER3 and POS3 pronoun is assigned an antecedent, the P_{ia} and R_{ia} figures are virtually identical; thus, in effect, under the current configuration of ROSANA-NN, they boil down to the canonical accuracy measure as described above in section 3.2.[21] As it turned out that the relative performance ranking of the different settings in the *ia* discipline is identical with the performance ranking in the non-pronominal anchors (*na*) discipline, the (P_{na}, R_{na}) tradeoffs are not depicted at this stage of consideration. In fact, the (P_{ia}, R_{ia}) results should be regarded the proper base of comparison as they immediately capture the extrinsic classifier performance, while the *na* results are sensitive to error chaining effects that should not be ascribed to the classifiers.

Notably and somewhat unexpectedly, it is the classifiers' cumulated (C ∪ N) accuracy that seems to be of higher relevance. According to the determined weighted average results, the extrinsically best performing classifiers correspond to the settings *a*, *d* (PER3), *D*, and *A* (POS3), and are thus the very classifiers that intrinsically score highest on the cumulated (C ∪ N) set of cases (see figure 5). Concerning possessive pronouns, it is interesting to see that setting *D* rather

possessives; likewise, for evaluating the possessive settings *A*, *B*, *C*, *D*, the classifier pertaining to setting *a* has been chosen as the *standard classifier for non-possessives.*

[21] Document set ds_4 gives rise to a single exception as it contains an instance of the non-possessive pronoun "*them*" occurring near the beginning of a document for which an antecedent fulfilling the congruence constraint could not be found because the single correct *pl* candidate was erroneously assigned the numerus attribute *sg*.

than A seems to be the clear extrinsic winner, hence giving evidence that, as suspected above, data distribution is indeed an issue, making the D classifier scoring extrinsically higher than the A classifier despite the fact that the latter quantitatively outperforms the former at intrinsic level.

			antecedents (P_{ia}, R_{ia})		anchors (P_{na}, R_{na})	
System	Setting	Corpus	PER3	POS3	PER3	POS3
ROSANA-NN	*(a,D)*	$cv_6(d_1^{53})$	$(0.64, 0.64)$	$(0.74, 0.74)$	$(0.61, 0.61)$	$(0.64, 0.64)$
ROSANA-ML	$(1^{tc}_{nc}$,h)	$cv_6(d_1^{66})$	$(0.66, 0.66)$	$(0.75, 0.75)$	$(0.62, 0.62)$	$(0.68, 0.68)$
	$(1^{tc}_{nc}$,h)	$[d_1^{31}, d_{32}^{66}]$	$(0.65, 0.64)$	$(0.76, 0.76)$	$(0.62, 0.61)$	$(0.73, 0.73)$
ROSANA	standard	$[d_1^{31}, d_{32}^{66}]$	$(0.71, 0.71)$	$(0.76, 0.76)$	$(0.68, 0.67)$	$(0.66, 0.66)$

Fig. 8. anaphor resolution results: ROSANA-NN vs. ROSANA-ML and ROSANA

If one thus combines the extrinsically highest-scoring classifiers for non-possessives and possessives, viz., the PER3 classifier corresponding to setting a, and the POS3 classifier corresponding to setting D, the cumulated and averaged extrinsic cross-validation results shown in figure 8 are obtained for ROSANA-NN. The (P_{ia}, R_{ia}) results of $(0.74, 0.74)$ shown for POS3 stem from the very experiment that has given rise to the results shown in column D of figure 7. The results of $(0.64, 0.64)$ for PER3 as well remain identical to the results in column a of figure 6. Furthermore, figure 8 gives the results in the non-pronominal anchors (na) evaluation discipline. Due to error chaining, it is, in general, harder to determine a cospecifying non-pronominal antecedent than an arbitrary antecedent; hence, the (P_{na}, R_{na}) tradeoffs of $(0.61, 0.61)$ for non-possessives and $(0.64, 0.64)$ for possessives are lower than the respective (P_{ia}, R_{ia}) tradeoffs.[22]

5 Comparison

Evaluation results of ROSANA-NN's ancestors are included in figure 8.[23] At first glance, ROSANA-NN seems to perform slightly worse than ROSANA-ML if one takes the *immediate antecedency* tradeoffs as the base of comparison.[24]

[22] See [5] for a more elaborate discussion of this issue.

[23] For a proper interpretation of these figures, one should take into account that the employed evaluation corpora differ slightly: $cv_6(d_1^{53})$ refers to the extrinsic cross-validation on the above-considered redundancy-free corpus of 53 news agency press releases; $cv_6(d_1^{66})$ refers to the 6-fold extrinsic cross-validation on the full - to a certain extent redundant - set of 66 news agency press releases as employed for cross-validating ROSANA-ML; $[d_1^{31}, d_{32}^{66}]$ refers to the bipartition of the full set of 66 press releases into a development vs. an evaluation corpus as employed for developing and assessing the original ROSANA system.

[24] This is justified as the (P_{ia}, R_{ia}) tradeoffs immediately capture the extrinsic performance, while the (P_{na}, R_{na}) results are sensitive to error chaining effects - see the above discussion.

Importantly, however, it should be taken into account that the ROSANA-NN evaluation data stem from experiments on a redundancy-free corpus, whereas the ROSANA-ML results have been obtained on a corpus exhibiting a certain degree of redundancy, which might be suspected to facilitate the learning task. Thus, given that a more difficult corpus has been employed, the results indicate that ROSANA-NN performs at least similar to its decision-tree-based ancestor.

Compared with its salience-based ancestor ROSANA, ROSANA-NN performs comparably on possessives, whereas it lags significantly behind on non-possessives. This confirms the findings of the C4.5 decision tree experiments (see [10]), according to which non-possessives are harder to deal with by ML means than possessives. The inferior results on non-possessives might be taken as an indicator that still the required sources of evidence are not adequately captured, and that the inventory of features over which the signatures are defined should be appropriately refined.

Finally, comparing the outcomes of the ROSANA-NN evaluation with the results given by Connolly et al. ([6]), it has to be taken into account that they consider the harder pronoun resolution task of determining *non-pronominal* antecedents. According to their findings, the two investigated types of backpropagation networks score highest, achieving an extrinsic accuracy of 0.55 (subspace-trained backpropagation networks) and 0.52 (ordinary backpropagation networks) for pronouns; no distinction between non-possessives and possessives is drawn. While a proper comparison should be based on a common evaluation corpus, this might be interpreted as a first indicator that the neural-network-based anaphor resolver considered above performs better, as its cumulated non-possessive $\cup$ possessive accuracy regarding non-pronominal antecedents amounts to 0.62.

6 Conclusion

Taking the previous work on the manually knowledge-engineered anaphor resolution system ROSANA and its successful hybrid, partly decision-tree-based descendant ROSANA-ML as the points of departure, it has been investigated what can be gained by employing backpropagation networks as the machine learning device for automatically determining antecedent preference criteria for pronoun resolution, leaving the filtering criteria to the discretion of the knowledge engineer. This research was motivated by the findings of Connolly et al. ([6]), who gained evidence that, compared to C4.5 decision trees, the standard backpropagation algorithm might be slightly ahead with respect to the task of object (NP) anaphor resolution.

According to the above results, the hybrid neural network approach ROSANA-NN performs similar to its decision-tree-based ancestor ROSANA-ML; given that a more difficult corpus has been employed for evaluation, it might even be the case that ROSANA-NN is slightly ahead. Extrinsic cross-validation on a corpus of 53 press releases has shown that ROSANA-NN achieves an accuracy of 0.64 on third-person non-possessives and of 0.74 on third-person possessives in the evaluation discipline of immediate antecedency. Thus, while these results do not yet allow to conclude that backpropagation networks are the unique best choice, they at least indicate that backpropagation networks are among the most successful machine learning model for anaphor resolution, in as far supporting the findings of Connolly et al. ([6]).

The methodology for systematically dealing with the numerous experimental degrees of freedom regarding the application of backpropagation network classifiers to anaphor resolution constitutes itself a valuable contribution. A two-stage approach has been developed in which signature optimization issues are confined to stage 1, and intrinsic as well as extrinsic cross-validation are confined to stage 2. Overall evaluation efforts thus remain bearable.

Subsequent research should address the question whether the above results generalize to larger text corpora and other text genres. Moreover, it should be instructive to investigate subspace-trained backpropagation networks, as this is the machine learning model that Connolly et al. ([6]) found to perform even better. Furthermore, there are recent advances in the field of rule learning, e. g. the SLIPPER approach by Singer and Cohen ([19]), which have not been taken into account in the earlier work of Connolly et al. ([6]); it should thus be worthwhile to consider these models as alternatives to neural network and decision tree learning. Regarding the ROSANA-NN algorithm itself, further experiments should address the question whether, by referring to the actually real-valued classification outcome ε, it can be suitably biased towards high precision; respective experiments have proven successful under the employment of decision trees (see [10]).[25] Finally, it should be revealing to compare ROSANA-NN, ROSANA-ML, and ROSANA based on a detailed qualitative analysis of their respective so-called competence cases according to the framework proposed in [20].

References

1. Lappin, S., Leass, H.J.: An algorithm for pronominal anaphora resolution. Computational Linguistics **20**(4) (1994) 535–561
2. Kennedy, C., Boguraev, B.: Anaphora for everyone: Pronominal anaphora resolution without a parser. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING). (1996) 113–118
3. Baldwin, B.: Cogniac: High precision coreference with limited knowledge and linguistic resources. In Mitkov, R., Boguraev, B., eds.: Proceedings of the ACL'97/EACL'97 Workshop on Operational Factors in Practical, Robust Anaphor Resolution for Unrestricted Texts, Madrid. (1997) 38–45
4. Mitkov, R.: Robust pronoun resolution with limited knowledge. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING'98/ACL'98), Montreal. (1998) 869–875
5. Stuckardt, R.: Design and enhanced evaluation of a robust anaphor resolution algorithm. Computational Linguistics **27**(4) (2001) 479–506

[25] Biasing towards high precision is deemed important for typical applications of anaphor and coreference resolution such as text summarization and question answering.

6. Connolly, D., Burger, J.D., Day, D.S.: A machine-learning approach to anaphoric reference. In: Proceedings of the International Conference on New Methods in Language Processing (NEMLAP). (1994)
7. Aone, C., Bennett, S.W.: Evaluating automated and manual acquisition of anaphora resolution strategies. In: Proceedings of the 33rd Annual Meeting of the ACL, Santa Cruz, New Mexico. (1995) 122–129
8. Ge, N., Hale, J., Charniak, E.: A statistical approach to anaphora resolution. In: Proceedings of the Sixth Workshop on Very Large Corpora, Montreal. (1998) 161–170
9. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics **27**(4) (2001) 521–544
10. Stuckardt, R.: A machine learning approach to preference strategies for anaphor resolution. In Branco, A., McEnery, T., Mitkov, R., eds.: Anaphora Processing: Linguistic, Cognitive, and Computational Modelling, John Benjamins (2005) 47–72
11. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the ACL, Philadelphia. (2002) 104–110
12. Ng, V., Cardie, C.: Weakly supervised natural language learning without redundant views. In: HLT-NAACL 2003: Proceedings of the Main Conference. (2003) 173–180
13. Olsson, F.: A survey of machine learning for reference resolution in textual discourse. SICS Technical Report T2004:02, Swedish Institute of Computer Science (2004)
14. Grüning, A., Kibrik, A.A.: Modelling referential choice in discourse: A cognitive calculative approach and a neural network approach. In Branco, A., McEnery, T., Mitkov, R., eds.: Anaphora Processing: Linguistic, Cognitive, and Computational Modelling, John Benjamins (2005) 163–198
15. Mitkov, R.: Factors in anaphora resolution: They are not the only things that matter. a case study based on two different approaches. In Mitkov, R., Boguraev, B., eds.: Proceedings of the ACL'97/EACL'97 Workshop on Operational Factors in Practical, Robust Anaphor Resolution for Unrestricted Texts, Madrid. (1997) 14–21
16. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
17. Järvinen, T., Tapanainen, P.: A dependency parser for english. Technical Report TR-1, Department of General Linguistics, University of Helsinki (1997)
18. Chomsky, N.: Lectures on Government and Binding. Foris Publications, Dordrecht (1981)
19. Cohen, W.W., Singer, Y.: A simple, fast, and effective rule learner. In: Proceedings of the 16th National Conference on Artificial Intelligence (AAAI), Menlo Park, CA. (1999) 335–342
20. Stuckardt, R.: Three algorithms for competence-oriented anaphor resolution. In: Proceedings of the 5th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 04). (2004) 157–163