Enhancing Coreference Clustering

Manfred Klenner
Computational Linguistics
Zurich University
Switzerland,
klenner@cl.uzh.ch

Étienne Ailloud
Computational Linguistics
Zurich University
Switzerland,
ailloud@cl.uzh.ch

Abstract

We introduce a new approach for the generation of coreference chains from pairwise classified markables. Depending on the experimental setting, we achieve an F-measure improvement between 2 and 6 percent. Coreference clustering is formulated as a minimisation task. The costs are derived from a binary classifier. The integration of an anaphor candidate pair into the evolving coreference sets is restricted by consistency preserving constraints.

1 Introduction

Pairwise classification of anaphora candidate pairs is a straightforward and successful method. Its advantages are: feature selection is well explored, training and testing are fast. Depending on the evaluation scenario, the F-measure values center around 60–65% (all mentions considered, all phenomena covered) and 75–80% (true mentions only). However, the solution produced by such a pairwise classifier does not necessarily fulfil fundamental linguistic restrictions, for instance the transitivity of the anaphoric relation. This shortcoming stems from the strictly local perspective of pairwise classification: pairs of markables are classified independently of each other. Already carried out classifications do not influence current decisions. As a result, the set of implicitly generated coreference chains is bound to be inconsistent. For example, a number of demo systems on the Internet produce the coreference set {He_i, he_k, him_l} given the sentence 'He_i believes him_j, but he_k never trusts him_l'. Although these systems actually do adhere to intra-sentential binding constraints as can be seen from simpler examples where no inconsistencies are generated (e.g. 'He_i believes $\text{him}_{j\neq i}$.'), they fail to propagate exclusiveness. The key is the transitivity of the anaphoric relation. Since he_k and him_l are exclusive, if He_i and he_k are assumed to be coreferent, then He_i and him_l must be exclusive as well. A few approaches explicitly try to incorporate transitivity in their models. McCallum and Wellner (2005) and Culotta et al. (2007) regard anaphora resolution as a Relational Learning problem. Klenner (2007), and more recently Finkel and Manning (2008) propose an Integer Linear Programming (ILP) solution. Both approaches are burdened with a high computational complexity, ILP is even NP-complete.

We propose a model that is far less complex and at the same time adheres to exclusiveness and other compatibility restrictions as 'global constraints'. Our approach can be seen as coreference clustering. Coreference clustering is the process of gathering together the coreference sets from the output of a pairwise classifier. Several approaches have been proposed: closed-first, best-first and aggressive-merge. Although these heuristics might reduce inconsistencies (by chance), they do not form a systematic and theoretically satisfying solution. We propose an approach to coreference clustering that is inspired by the algorithm of Balas (Balas et al., 1965). The approach turns coreference clustering into a minimization task where coreference sets are incrementally generated starting from the safest pairs.

We first describe our gold standard data. Next we discuss the baseline classifier being used. We then introduce our model, discuss some of its properties and give an empirical evaluation.

2 Gold Standard Data

As a gold standard we use the TüBa-D/Z (Naumann, 2006; Telljohann et al., 2005) coreference corpus. The TüBa-D/Z is a treebank (1100

Johansson, C. (Ed.)
Proceedings of the Second Workshop
on Anaphora Resolution (2008)

German newspaper texts, 25,000 sentences) augmented with coreference annotations. In total, there are 13,818 anaphoric, 1031 cataphoric and 12,752 coreferential relations. The annotated data comprises 3295 relative pronouns, 131 attributive relative pronouns, 8929 personal pronouns, 2987 reflexive pronouns, and 3921 possessive pronouns.

Expletive 'es' (a non-referential 'it', e.g. 'it rains') is rather frequent: 2017 out of 2231 occurrences are expletive. So we decided not to consider 'es'. 791 of the (remaining) pronouns are not annotated as being anaphoric. Some of them actually are non-anaphoric[1]. Non-anaphoric use often comes as a first person plural pronoun (e.g. 340 cases of German 'wir' = English 'we'). They typically occur in newspapers in statements as: 'the government never told us (us = the citizen) that ...'. Here, 'us' appears only once and without any actual reference anchored within the text.

3 Baseline Classifier

We use TiMBL (Daelemans et al., 2007), a memory-based (lazy) learner, as a classifier. We have experimented with various features for the training and test vectors, e.g. distance in markables or a binary feature indicating whether two markables have the same grammatical function (parallelism). Table 1 lists the complete feature set used in the experiments described below.

- distance in sentences
- distance in markables
- part of speech of the head of the markables
- the grammatical functions
- parallelism of grammatical functions
- do the heads match or not
- where is the pronoun (if any): left or right
- wordform if POS is pronoun
- salience of the non-pronominal phrases

Table 1: Features

Our *pair generation component* restricts the candidate pairs to those that morphologically agree, and it guarantees that exclusiveness restrictions from binding theory are not violated. Binding constraints require e.g. that a pronoun is free, i.e. must not be c-commanded by a coreferent noun phrase. We have defined two principles that approximate the c-command. See section 5 for a discussion.

The classifier expresses morphological agreement of the candidate pairs as a hard filter: The vector for a pair is generated only if agreement is successful. For instance, in German, two personal pronouns must agree in person, number and gender, while a possessive pronoun must only bear the same person and gender as its antecedent ('sie$_i^{sing}$ ..', 'ihren$_i^{plur}$ Kindern^{plur} ..')[2].

Nominal anaphor and antecedent must only agree in number, gender might be different ('Der Weg$_i^{masc}$ ..', 'Die Strecke$_i^{fem}$ ..').

Since we currently do not have any semantic features, we deal only with those nominal anaphora where both markables (partially) match: e.g., 'Woody Allen', 'Mr. Allen' etc., but not 'Mr. Allen' and 'The comedian'. We excluded these about 1000 'pure' nominal anaphora from our evaluation.

Table 2 shows the informal definition of a *valid candidate pair*, where m_i and m_j are markables. Only valid candidate pairs will be generated for pairwise classification.

A pair $\langle m_i, m_j \rangle$ is a *valid candidate pair* if

- it adheres to the (intra-sentential) binding constraints
- m_i and m_j agree morphologically
- m_i and m_j are semantically compatible (here: they match).

Table 2: Valid Candidate Pair

There are some rather long texts in the Tüba corpus. Which pair generation algorithm is reasonable? Should we pair every markable (even from the beginning of the text) with every other markable (till the end of text)? This is neither computationally feasible nor linguistically plausible. For example, pronouns are acting as a

[1] We have found some cases where an anaphoric pronoun has not been annotated. Currently, we check all 791 cases in order to find out how reliable these annotations are.

[2] In German, a possessive pronoun and its head agree in number.

For animates, grammatical and natural gender might disagree between a pronoun and its antedecent ('Das Mädchen$_i^{neut}$ ..', 'sie$_i^{fem}$'—'the girl$_i$ ..', 'she$_i$'), but the phenomenon is rare enough in the data to ignore it.

ISSN 1736-6305 Vol. 2
http://dspace.utlib.ee/dspace
/handle/10062/7129

kind of local variables. A 'he' at the beginning of a text and a second distant 'he' at the end of the text hardly tend to co-refer, except if there is a long chain of coreference 'renewals' that lead somehow from the first 'he' to the second 'he'. But the plain 'he'-'he' pair does not reliably indicate coreference.

A smaller window, for example 3 sentences, seems to be more appropriate. We generate candidate pairs only within that window, which is moved sentence-wise over the whole text. Alternatively, the algorithm proposed by (Soon et al., 2001) could be used.

Please note that one of our central claims is that any kind of pairwise classification, window-based or not, inadvertently produces inconsistent coreference chains (see section 6 for a discussion). So our approach to coreference clustering (the process of making the implicit partition explicit) has to assure that coreference sets only contain markables that form valid pairs (in the sense defined above).

4 Clustering Model

The output of the pairwise classifier serves as input to our system. It takes all (but only the) positively classified pairs and orders them according to the *classification strength* of TiMBL's decision. The classification strength is derived from the number of positive and negative instances that TiMBL has found to be most similar to the candidate pair in question. Our classification cost measure $w_{i,j}$ is defined by

$$w_{i,j} = \begin{cases} 0 & if \mid neg_{i,j} \mid = 0 \\ \frac{\mid neg_{i,j} \mid}{\mid neg_{i,j} \cup pos_{i,j} \mid} & else \end{cases},$$

where i and j are markable indices (the markables are linearly ordered from left to right and receive their position as index). For convenience, we denote a weighted candidate pair as $\langle w_{i,j}, m_i, m_j \rangle$. $neg_{i,j}$ and $pos_{i,j}$ are the sets of (the most similar) positive and (the most similar) negative instances, respectively. Our model seeks to minimise classification costs, inferred from the number of counterexamples to TiMBL's decisions. Thus, if no negative instance is found, it proposes a safe positive classification decision at zero cost. Accordingly, the cost of a decision without any positive instances is high, namely one. Otherwise, the ratio of the negative instances to the total of all instances is taken[3]. For example, if TiMBL finds 10 positive and 5 negative examples the cost of a positive classification is 5/15 while a negative classification costs 10/15.

forall $\langle w_{i,j}, m_i, m_j \rangle \in OS_{w_{i,j} \leq 0.5}$
1 forall $\langle S_k, P_k \rangle \in SVC$
2 $SVC = SVC \setminus \{\langle S_k, P_k \rangle\}$
3 choose $C_i \in P_k$ such that $m_i \in C_i$
(if $C_i = \emptyset$ then $C_i = \{m_i\}$)
4 choose $C_j \in P_k$ such that $m_j \in C_j$
(if $C_j = \emptyset$ then $C_j = \{m_j\}$)
5 if forall $m_u \in C_i$ and $m_v \in C_j$: valid(m_u, m_v)
6 $C_{i,j} = C_i \cup C_j$
7 $S_{k1} = S_k + w_{i,j}$
8 $P_{k1} = P_k \setminus \{C_i, C_j\}$
9 $P_{k1} = P_{k1} \cup \{C_{i,j}\}$
10 $SVC = SVC \cup \{\langle S_{k1}, P_{k1} \rangle\}$
11 $S_{k0} = S_k + 1 - w_{i,j}$
12 $SVC = SVC \cup \{\langle S_{k0}, P_k \rangle\}$
13 SVC = prune(SVC,N-best)
return argmin$_P\{S; \langle S, P \rangle \in SVC\}$

Figure 1: Coreference Clustering

Fig. 1 shows the core of the clustering algorithm. Let $OS_{w_{i,j} \leq 0.5}$ be the ordered set of candidate pairs with a cost $w_{i,j}$ less than or equal to 0.5. The first element of $OS_{w_{i,j} \leq 0.5}$ represents the strongest choice, since it has the lowest cost. Note that such a cost-based ordering does not preserve (textual) linearity. Exactly this is a crucial property of our incremental approach to coreference clustering: that it bases its actions on such strong decisions. Actually, these define reliable cluster seeds (we evaluate our ordering principle in section 9.2).

Coreference clustering is carried out by an n-best beam search with pruning. The vector set SVC defines the beam. It consists of pairs of the form $\langle S_k, P_k \rangle$, where S_k is the accumulated cost of the whole partition of coreference sets $P_k = \{.., \{.., m_i, ..\}, \{.., m_j, ..\}, ..\}$.

Given a candidate pair $\langle m_i, m_j \rangle$ and its weight $w_{i,j}$, every $\langle S_k, P_k \rangle$ is expanded into at most two successor versions. One version where

[3] TiMBL's default IB1 algorithm actually inspects k-nearest-*distances* and thus never fails to return instance candidates.

m_i and m_j are assumed to be coreferent (line 5 to 10) and one version where the pair is discarded (line 11 and 12). m_i and m_j are taken as coreferent only if their coreference sets, C_i and C_j, may be merged (line 5). Two coreference sets are mergeable if each pair contained in the Cartesian product $C_i \times C_j$ forms a *valid pair*. See Table 2 for the definition of *valid pair*.

If all pairs are *valid*, then m_i and m_j are deemed coreferent (in this branch) and their coreference sets actually are merged (line 6). The score S_{k1} of that beam is determined by the costs so far accumulated, S_k, plus the weight of $\langle m_i, m_j \rangle$, namely $w_{i,j}$ (line 7). Finally, the new partition P_{k1} is generated: the coreference sets C_i and C_j are deleted and the merged set $C_{i,j}$ is added (lines 8 and 9).

We cannot rule out at this stage of the expansion that in subsequent derivations a version of P_k where m_i and m_j are not interpreted as coreferent proves to be superior to the partition where they are taken to be coreferent. So a second version of $\langle S_k, P_k \rangle$ is generated with an unaltered P_k. The cost of this decision is given by $1 - w_{i,j}$, i.e., $S_{k0} = S_k + 1 - w_{i,j}$ (line 11).

If all elements of SVC have been augmented for a given pair $\langle m_i, m_j \rangle$, SVC is ordered according to the costs and pruned (line 13)[4].

Clustering stops if all pairs from $OS_{w_{i,j}}$ have been processed this way. The algorithm returns the partition P with the lowest cost S. It represents the best coreference partition found (but not necessarily the globally optimal one, since we are pruning).

We carried out a single experiment to find out to what extent the solution with the lowest costs is at the same time the best out of the solutions of the beam. We considered 992 texts (N-best was set to 16). In 477 cases, the solution with the lowest score actually was the best from the beam. For the remaining cases, we measured the distance in F-measure values from the best solution to the highest ranked (i.e. with lowest costs). The average of these differences represents the gain in F-measure values one could win if the best solution out of the beam could be (somehow) reliably identified. It is 8.6%. Although this would represent a significant improvement, it is unclear how to achieve it.

[4]We have experimented with N-best = 4,8,16,32; our experiments are based on N-best = 16.

5 Binding Constraints

Although we are working with a treebank, i.e. (almost) perfect syntactic structures, we do not want to adhere to the c-command in its strictest form. The reason is that we want to replace the treebank (once) by the output of a statistical (dependency) parser. Any c-command reconstruction would be error prone. We rather defined two easily derivable predicates that replace the c-command (and approximate it): *clause_bound* and *np_bound*.

Two mentions, m_i and m_j, are *clause-bound*, if they occur in the same subclause, none of them being a reflexive or a possessive pronoun, and they do not form an apposition. There are only 16 cases in our data set where this predicate produces false negatives. Some of these cases are country names reoccuring in the same clause as part of an adjectival phrase ("Russia$_i$ and Russian$_i$ people ..."). False negatives might also stem from clauses with predicative verbs ("He$_i$ is still prime minister$_i$"), where as an effect of the predicative construction both NPs are annotated in the same coreference set.

Two markables m_i, m_j that are clause-bound (in the sense defined above) are exclusive.

A possessive pronoun is exclusive to all markables in the (base) noun phrase it is contained in (e.g. "[her$_i$ manager$_j$]" with $i \neq j$), but might get coindexed with markables outside of such a local context ("Anne$_i$ talks to her$_i$ manager"). We define a predicate *np_bound* that is true of two markables m_i and m_j if they occur in the same (base) noun phrase. In general, two markables that *np-bind* each other are exclusive.

6 Model Properties

Our clustering model seeks to establish consistent coreference sets while optimising the solution. It enforces consistency requirements via hard constraints such as those used in our definition of a valid pair. There are two main sources for the inconsistency as introduced by pairwise classifiers:

(a) exclusiveness restrictions (binding constraints) propagate (through transitivity), but pairwise classifiers have a limited perspective;

ISSN 1736-6305 Vol. 2
http://dspace.utlib.ee/dspace
/handle/10062/7129

(b) underspecified words (either by their lexical nature or as a result of a shallow processing) are bridges for inconsistent pairs.

To illustrate (b), consider e.g. German 'sich' (reflexive pronoun). It has unspecified gender and number values. TiMBL might classify candidate pairs such as (er,sich) and (sie,sich)[5] as both positive, although they are implicitly incompatible. Other underspecified word classes are named entities whose (grammatical) gender is often unknown (surnames, product names, even cities etc. may have a non-neutral grammatical gender in German). If the gender is unknown both pairs (Berlin, seine) and (Berlin, ihre)[6] might be classified as positive, although 'seine' (his) und 'ihre' (her) are incompatible.

But the need for constraint checking while clustering more fundamentally stems from the pair generation algorithm itself, namely the look-up window (most approaches define such a window – to reduce the amount of negative instances in order to prevent the classifier from generating a bias). In augmenting coreference sets with new elements, pairs that never have been considered for pairwise classification (because they are outside the window) must be verified; e.g., two nouns must be evaluated wrt. their semantic compatibility.

So the source of confusion is high and the potential of improvements as well. Our approach works by removing false positives, so to speak. The removal is a side effect of coreference clustering, forced by consistency checks and guided by optimisation.

7 Entity-Constrained Measure

A traditional evaluation scheme for coreference resolution has been that of the Message Understanding Conference task:the link-based measure from (Vilain et al., 1995). While it has been widely used, and is simple to implement (basically a ratio of common links between key and response to the whole number of links) the MUC measure fails to grasp some distinctions between outputs of coreference classification. Being based on coreference links, it makes no difference between wrongly classified links of distinct nature: For instance, it punishes with the same factor the wrong merging of any two coreference sets, independently of the size of the merged set (see (Bagga and Baldwin, 1998) for a more comprehensive discussion of these issues).

The Entity-Constrained Measure (ECM) is introduced in (Luo et al., 2004) and rebaptised entity-based Constraint Entity-Alignment F-measure (CEAF) in (Luo, 2005). It addresses some of the shortcomings of the MUC measure, in that it relies on similarities between whole *sets* ("entities") of coreferent markables in order to align key and response coreference partitions. The ratio of thus successfully aligned markables to the total number of key (resp. response) markables yields recall (resp. precision) for the ECM measure.

Since our algorithm focusses on coreference chains rather than linkages of markables, ECM seems a better choice than MUC to correctly assess the gain of accuracy induced by, say, global consistency constraints. It is here the quality of the coreference chains that is stressed, which is well rendered by the similarity-based alignments from ECM.

As an illustration, the example from Table 3 displays a configuration where ECM is better suited than MUC: Assuming a gold standard of 5 mentions spread in 2 coreference sets $\{m_1, m_2, m_3\}$ and $\{m_4, m_5\}$, a baseline classifier would produce, say, a unique coreference set $\{m_1, m_2, m_3, m_4, m_5\}$. If one assumes that m_3 and m_5 are inconsistent, the implicit clustering of pairwise ML decisions will have overlooked it. Now a typical effect of our algorithm, basing on m_3 and m_5's not forming a valid pair, would be to split this coreference set into, say, $\{m_2, m_3\}$ and $\{m_1, m_4, m_5\}$.

However, as for MUC, this would mean a better partition via the baseline classifier (F-measure 0.857 vs. 0.667). On the contrary, the ECM measure prefers the split partition $\{\{m_1, m_2, m_3\}, \{m_4, m_5\}\}$ output by reclustering (0.8 vs. 0.6), without even decreasing recall. This is due to our algorithm's being able to align more markables with the reference partition after reclustering (m_2, m_3, m_4 and m_5 vs. only m_1, m_2 and m_3 for the baseline). This suits better our purposes of yielding higher-quality coreference chains, not least because our algorithm reduces the number of linkages between markables, which tends to be penalised by MUC.

[5] (he,himself) and (she,herself)
[6] (Berlin, his) and (Berlin, her)

	C_1	C_2	MUC			ECM		
Reference	$\{m_1, m_2, m_3\}$	$\{m_4, m_5\}$	P	R	F	P	R	F
Classifier	$\{m_1, m_2, m_3, m_4, m_5\}$		3/4	3/3	0.857	3/5	3/5	0.6
Own	$\{m_2, m_3\}$	$\{m_1, m_4, m_5\}$	2/3	2/3	0.667	4/5	4/5	0.8
reclustering effect					drop			boost

Table 3: Illustration: MUC score compared to ECM score

8 Related Work

As in our algorithm, some other approaches rely on smart exploration of the search space for coreference among markables.

In (Luo et al., 2004), the algorithm performs the search by incrementally constructing the Bell tree, which represents the possible partitions into markable sets (entities). Paths in the tree are scored against a maximum-entropy model that has been trained on entity-wide coreference information. The most notable differences with our approach lie in their monolithic learning phase: Traversal of the Bell tree only triggers pruning according to scoring threshold for the current branch; that is, virtually all linguistic information is digested during learning already. A subsequent difference is also that our algorithm does not perform linear left-to-right processing of a text: It rather starts from the most probable pairs – already pre-classified, then spreads on the global scale to check their validity.

Denis and Baldridge (Denis and Baldridge, 2007) use ILP to jointly determine anaphoricity and coreference. However, they do not use any propagating constraint, like transitivity of the coreference relation, to consolidate their output.

Work on anaphora resolution for German on the basis of the Tüba coreference corpus was carried out by (Hinrichs et al., 2005) and (Versley, 2006). A direct comparison with their results is difficult for several reasons. First, the work described in (Hinrichs et al., 2005) is restricted to pronominal anaphora resolution (mostly third person personal and possessive pronouns). Moreover, they used a former (smaller) version of the Tüba corpus. (Versley, 2006), on the other hand, concentrates on nominal anaphora exclusively – something that we currently are able to model only in part.

The clustering of the coreference pairs output by the classifier constitutes one of the steps in machine learning approaches to coreference resolution that may be tuned according to the needs of the experiment. As well as feature selection is an important decision, or sometimes the choice of a suitable classifying algorithm, coreference clustering decides on which final information will be available to the evaluation module, or, ideally, to the back-end application utilising coreference information.

(Ng, 2005), for instance, introduces a model in which the best combination of each of three decisions is learned, so as to produce the best possible results for different design choices in the ML system. The three clustering schemes there introduced all rely on finding a suitable antecedent for a given anaphor. The anaphor is considered from left to right, its potential antecedent from right to left. The schemes are: **closest-first**, where the first preceding markable that is coreferent is chosen; **best-first**, where the antecedent is chosen via likelihood among a fixed-size set of preceding coreferent markables; and **aggressive-merge**, where *all* preceding markables that are coreferent shall belong to the coreference set. The first two ones induce the coreference chains implicitly, by incrementally expanding the separate anaphor-antecedent pairs into coreference sets.

While learning to choose among various approaches is an interesting approach, it does not improve quality of the clustering process itself as our approach does.

Cardie and Wagstaff (Cardie and Wagstaff, 1999) introduced noun phrase coreference as clustering. They defined a distance measure based on linguistic and positional features,e.g. gender, head noun and semantic class of a noun phrase, their relative position and the distance between the candidates. Most of these features have been used (before and afterwards) in systems that carry out pairwise classification. The main difference of course is that (Cardie and Wagstaff, 1999) has an unsupervised approach while most pairwise anaphora resolution sys-

ISSN 1736-6305 Vol. 2
http://dspace.utlib.ee/dspace
/handle/10062/7129

tems go through a training phase and thus are supervised. However, their model is not fully unsupervised: they introduce a clustering radius which needs to be empirically adjusted (by a human). Also, they set the weights manually (that is, heuristically). Our system is fully empirically based, no manual tuning is necessary. Our weights stem from a pairwise classifier and our clustering algorithm uses a n-best beam search in order to approximate the search for an optimal solution. In (Cardie and Wagstaff, 1999), the algorithm performs a 'single shot' right-to-left search, while our beam search is guided by a low cost ordering of candidate pairs (we have shown that the accuracy of the clustering depends on such an ordering). Finally, we use an important linguistic constraint that (Cardie and Wagstaff, 1999) omits, namely intra-sentential binding constraints.

The same is true of the clustering approach described in (Yang et al., 2004). It is a supervised one, that includes information about whole chains of coreference, and therefore also addresses our main claim that binary information only is not enough in machine learning approaches. The model there incrementally builds the coreference chains by classifying the current noun phrase against every previously found chain, according to a decision tree of similarly learned instances. Yet there are no hard linguistic constraints as exclusiveness in this one either, since it relies on the sole machine learning phase, which is intertwined with the actual clustering, but which our model permits to revise in the subsequent reclustering.

9 Empirical Evaluation

We have carried out a series of experiments, most of them in order to evaluate or core model (false positive filtering), but we also have experimented with an extension of the core model, where we have tried to find and revert false negatives.

9.1 Experiments with the Core Model

Our experiments are based on a five-fold cross-validation setting (1100 texts from the Tüba coreference corpus). We carried out each experiment in two variants. One with all markables taken as input – an application-oriented setting, and one with only markables that represent true mentions (cf. (Ponzetto and Strube, 2006) and (Luo et al., 2004) for other approaches with an evaluation based on true mentions only). The assumption is that if one considers only true mentions, the effects of a model can be better measured.

The five test sets generated by TiMBL (the input to our clustering approach) consisted of 39,711 pairs in the 'true mention' setting and 138,740 pairs in the 'all mention' setting.

	all mentions			true mentions	
	Timbl	Own		Timbl	Own
setting I	63.98	65.09	F	76.89	78.74
excl.	67.60	71.22	P	75.91	78.33
	59.34	59.95	R	77.90	79.16
setting II	63.98	65.74	F	76.89	80.08
+ morph.	67.60	72.02	P	75.91	79.71
filter	60.75	60.49	R	77.90	80.45
setting III	63.98	65.69	F	76.89	82.50
+ NP	67.60	72.72	P	75.91	83.95
match	60.75	59.91	R	77.90	81.11

Table 4: Fishing for False Positives

Table 4 shows the ECM results of our experiments. We have used aggressive-merging to get the (ECM) baseline from TiMBL's output (labelled Timbl). Our system (label Own) takes TiMBL's positively classified pairs as input and derives weights from it in order to produce consistent and (quasi-)optimal partitions of coreference sets.

We have measured the relative influence of our three constraint types. In setting I, only exclusiveness was tracked, i.e. the definition of *valid pair* excluded the morphological filter as well as the matching restriction for non-pronominal noun phrases. The morphological filter was added in setting II. Finally, in setting III, the matching criteria for noun phrases was part of the definition.

Starting with the 'all mention series' of experiments, we can see from Table 4 that the exclusiveness constraint already contributes about 3.5% precision improvement (from 67.60 to 71.22). Since recall does not improve, the effect on the F-measure is less striking (about 1%). Setting II sees a 1.76% improved F-measure while setting III gives 1.71% (slightly worse, though). The gain in precision is about 5%. Note that this improvement stems from nothing but the propagation of constraints. No false

negatives have been turned into positives, only false positives have been removed (on consistency grounds and under the regiment of optimisation).

The effects become more drastic in the 'true mention series'. We get an improvement of 1.85%, 3.19% and 5.61% F-measure in setting I, II and III respectively. The gain in precision in setting III is almost 8%.

As mentioned above, our approach 'removes' only false positives while it generates coreference sets. Just to give an impression on the number of false positives that we are talking about, we give here values of a concrete run. In setting I ('true mention series') our method has produced 777 markable pairings less than TiMBL (i.e. they have been 'removed', so to speak). 592 of them actually have represented false positives, the remaining 285 have been true positives. Please note that the loss of true positives does not necessarily result in a reduction of recall, since the ECM recall is not measured primarily in terms of linkages (as the MUC is). Moving to setting II, our system suppressed 928 linkages, this time 730 were false positives (that is, rightly suppressed linkages), only 200 were true positives. Finally, in setting III, the system revised 1949 linkages as negative, 1399 of them were correctly 'removed'.

Please note that the improvement over the baseline is not an artefact of our baseline definition (aggressive-merging of the output of TiMBL). None of the other methods discussed in the literature (best-first, ..) alone can guarantee consistent partitions. So one may expect an improvement with any of these methods (as a baseline provider). We have experimented with best-first, but the baseline results were worse than those of the aggressive-merging strategy, so we ceased these experiments.

9.2 Evaluation of 'Balas Ordering'

Our clustering method is inspired by the Balas algorithm for Zero-One ILP (Balas et al., 1965). As a look in the literature on ILP for NLP (e.g. (Roth and Yih, 2004)) reveals, all ILP formalisations are of that special type. The Balas algorithm, although still NP-complete, offers an interesting, since very clever approach to overcome some of the flaws of ILP (e.g. the need for extensionalisation of constraints). It seems feasible to implement special purpose variants of it that capture constraints intensionally (e.g. transitivity). Our coreference clustering approach is such a special purpose implementation, however an incomplete one: We do not search for a global optimum, but for an approximation of it via n-best beam search.

One crucial question then is: does the 'Balas ordering' of binary variables (our pairs) prove superior to other orderings? In other words: does the preference ordering over pairs (and thus seed clusters) influence the quality of the partition?

	all mentions			true mentions	
	iBal	Own		iBal	Own
setting I	63.07	65.09	F	77.50	78.74
excl. &	68.89	71.22	P	77.13	78.33
	58.10	59.95	R	77.88	79.16
setting II	63.71	65.74	F	78.63	80.08
+ morph.	69.75	72.02	P	78.28	79.71
filter	58.65	60.49	R	78.99	80.45
setting III	63.81	65.69	F	78.44	82.50
+ NP	70.50	72.72	P	80.01	83.95
match	58.27	59.91	R	76.93	81.11

Table 5: Effect of 'Balas Ordering'

In each of the runs from Table 5 we have clustered according to the Balas scheme (label Own) but also with the reversed ordering of pairs (label iBal, i.e. most expensive pairs first). As we can see from Table 5 ordering matters. For the 'true mentions' series (setting III) the improvement with the Balas ordering (label Bal) is 5.61% (82.5-76.89), while that with the inverse Balas ordering is 1.55% (78.44-76.89). For the 'all mentions' series the inverse Balas (iBal) ordering even worsens the results.

9.3 Experiments Part II

Pairwise classifiers (including TiMBL) sometimes leave some pronouns unattached (depending on the training data and the derived statistical model). That is, there is no candidate pair with cost < 0.5 (or, in a probability framework, with probability > 0.5) for any of those pronouns. Please note that this is a desirable feature since – as mentioned before – quite a number of pronouns are used non-anaphorically.

However, there might be some false negatives among these unattached pronouns. We have carried out experiments to explore the poten-

ISSN 1736-6305 Vol. 2
http://dspace.utlib.ee/dspace
/handle/10062/7129

tial of switching false negatives (pronouns) into (true) positives.

The non-anaphoric use of pronouns is often restricted to special types of pronouns (cf. section 2). If we had a good 'non-anaphoric pronoun' classifier to detect these uses, how much would our clustering approach profit from it? We have simulated a perfect classifier, i.e., we removed the 792 non-anaphoric uses of pronouns from our corpus. Any unattached pronoun produced by TiMBL is now a false negative.

We augmented our algorithm in the following manner: the ordered input queue $OS_{w_{i,j} \leq 0.5}$ was extended to also integrate pairs with non-anaphoric pronouns ('non-anaphoric' according to TiMBL's classification). That is, provided $OS_{w_{i,j} > 0.5}$ for all j where j is a pronoun (index), at least one $w_{i,j}$ is selected to hold, i.e. the augmented algorithm forces a seemingly non-anaphoric pronoun to get an anaphoric interpretation.

	all mentions			true mentions	
	$\oplus$	$\ominus$		$\oplus$	$\ominus$
TiMBL	63.98	64.83	F	76.89	77.76
	67.60	68.79	P	75.91	77.01
	60.75	61.17	R	77.90	78.53
Own	65.84	66.87	F	82.10	83.45
	66.64	68.44	P	81.20	82.61
	65.09	65.39	R	83.02	84.32

Table 6: Fishing for False Negatives

In Table 6 again we present two series, 'all mentions' and 'true mentions'. $\oplus$ represents the case where the original corpus (all pronouns, including non-anaphoric ones) is clustered with our modified algorithm. Clearly, this produces false positives, since true non-anaphoric pronouns are forced to be anaphoric. In the $\ominus$ setting, all non-anaphoric uses are discarded from the corpus. So each line in a cell shows the gain in performance if we could safely identify non-anaphoric uses of pronouns. TiMBL pushes its results ('true mention' series) from 76.89% to 77.76% (F-measure), i.e. an improvement of 0.87%. The comparison within a row under $\ominus$ illustrates whether our clustering approach could also benefit from such a scenario. To continue our example, TiMBL produces 77.76% F-measure, our system 83.45 % ('true mentions' series). The relative gain at the level of our system is 1.35% (83.45 - 82.10). Since TiMBL already improved its result by 0.87% only 0.58% of the overall improvement (1.35%) stems from the clustering level. The conclusion is: even in the best case scenario (perfect classifier) there is only a small effect. Further investigations are necessary in order to find whether other types of false negatives could be turned into (true) positives with a greater impact.

10 Conclusion and Future Work

We have introduced an approach to coreference clustering that seeks to optimise coreference partitions according to a n-best beam search that selects good seed clusters (first) and safely augments it under the regiment of constraints to form consistent coreference sets.

The key insight of our approach is that (linguistic) constraints not only need to be applied (e.g., as hard filters) at the level of pairwise classification, but also (again) at the level of coreference clustering. Pairwise classifiers implicitly produce coreference chains and these chains are corrupted by underspecified markables that build bridges between (transitively) incompatible pairs.

As discussed previously, we have (temporarily) excluded so-called bridging anaphora ('Allen' .. 'The comedian') from consideration, since we currently have no means to evaluate them properly. Our clustering approach would simply remove them. One would need at least animacy, but even better, taxonomic information, to define a 'bridging constraint'. We currently are augmenting our data in order to cope with that phenomenon. Ontological learning in the sense of (Lech and de Smedt, 2007) might be another option.

Our model has an impact of 2% up to 6% (F-measure). We have also demonstrated that our ordering principle itself is a good choice. Although in sum our approach works fine, its performance wrt. the 'real setting' (all mentions) is not fully satisfying (2%). Also, we still have a simplified nominal resolution component.

Future work will thus strive to come to a complete and better model. One crucial step is the integration of semantic features. But also other constraints might prove helpful (e.g. reflexive pronouns are bound in the sentence they occur

in, any coreference set must have a least one non-pronominal anchor). Finally, we plan to compare our model with a corresponding ILP formalisation in order to find out how far our best solution is from the optimal one.

Acknowledgement The work described herein is partly funded by the Swiss National Science Fondation (grant 105211-118108).

References

Bagga, Admit and Breck Baldwin. 1998. Algorithms for scoring coreference chains. *Proceedings of the Linguistic Coreference Workshop at LREC '98*, pp. 563–566

Cardie, Claire and Kiri Wagstaff. 1999. Noun phrase coreference as clustering. *EMNLP '99*.

Balas, Egon, Fred Glover and Stanley Zionts. 1965. An additive algorithm for solving linear programs with zero-one variables. *Operations Research*, 13(4), pp. 517–549.

Culotta, Aron, Michael Wick and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. *NAACL HLT '07*, pp. 81–88.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot and Antal van den Bosch. 2007. TiMBL: Tilburg memory based learner, version 6.1, reference guide. ILK Research Group Technical Report no. 07-07.

Denis, Pascal and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. *HLT '07*.

Finkel, Jenny R. and Christopher Manning. 2008. Enforcing transitivity in coreference resolution. *ACL '08*.

Hinrichs, Erhard W., Katja Filippova and Holger Wunsch. 2005. A data-driven approach to pronominal anaphora resolution in German. *RANLP '05*.

Klenner, Manfred. 2007. Enforcing consistency on coreference sets. *RANLP '07*.

Lech, Christopher Till and Koenraad de Smedt. 2007. In Christer Johansson (ed.): Proceedings from the first Bergen Workshop on Anaphora Resolution (WAR I), Cambridge Scholars Publishing, UK.

Luo, Xiaoqiang. 2004. On coreference resolution performance metrics. *ACL '05*, pp. 157–164.

Luo, Xiaoqiang, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. *ACL '04*.

McCallum, Andrew and Ben Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. *Advances in Neural Information Processing Systems*, 17, pp. 905–912.

Naumann, Karin. 2006. Manual for the annotation of in-document referential relations. Tech. Report. Seminar für Sprachwissenschaft, Universität Tübingen.

Ng, Vincent. 2005. Machine learning for coreference resolution: from local classification to global ranking. *HLT '05*, pp. 25–32.

Ponzetto, Simone P. and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. *HLT-NAACL '06*, pp. 192–199.

Roth, Dan and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. *CoNLL '04*.

Soon, Wee Meng, Hwee Tou Ng and Daniel Chun Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4), pp. 521–544.

Telljohann, Heike, Erhard Hinrichs, Sandra Kübler and Heike Zinsmeister. 2005. Stylebook for the Tübingen treebank of written German (TüBa-D/Z). Tech. Report. Seminar für Sprachwissenschaft, Universität Tübingen.

Versley, Yannick. 2006. A constraint-based approach to noun phrase coreference resolution in German newspaper text. *Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS)*.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. *MUC6 '95*.

Yang, Xiaofeng, Jian Su, Guodong Zhou and Chew Lim Tan. 2004. An NP-cluster based approach to coreference resolution. *COLING '04*.