of –Coref for all relations. Thus while +Coref pays a price in precision for its improved recall, in many applications it may be a worthwhile tradeoff.

Though one might expect that errors in coreference would reduce precision of +Coref, such errors may be balanced by the need to use longer patterns in –Coref. These patterns often include error-prone wildcards which lead to a drop in precision. Patterns with multiple wildcards were also more likely to be removed as unreliable in manual pattern pruning, which may have harmed the recall of –Coref, while improving its precision.

5.3 Further Analysis

Our analysis thus far has focused on micro-reading which requires a system find all mentions of an instance relation – i,e, in our evaluation *OrgLeader(Apple, Steve Jobs)* might occur in as many as 20 different contexts. While –Coref performs poorly at micro-reading, it could still be effective for macro-reading, i.e. finding at least one instance of the relation *OrgLeader(Apple, Steve Jobs)*. As a rough measure of this, we also evaluated recall by counting the number of test instances for which at least one answer was found by the two systems. With this method, +Coref's recall is still higher for all but one relation type, although the gap between the systems narrows somewhat.

	+Coref	-Coref	#Test Instances
ORGEmploys	8	2	20
GPEEmploys	**12**	3	19
hasSibling	**11**	4	19
hasBirthDate	**12**	5	17
hasSpouse	**15**	2	20
ORGLeader	**14**	6	19
attendedSchool	**17**	12	20
hasBirthPlace	**19**	15	20
GPELeader	**15**	13	19
hasChild	6	6	19

Table 2: Number of test seeds where at least one instance is found in the evaluation.

In addition to our recall evaluation, we measured the number of sentences containing relation instances found by each of the systems when applied to 5,000 documents (see Table 3). For almost all relations, +Coref matches many more sentences, including finding more sentences for those relations for which it has higher precision.

6 Conclusion

	Prec		Number of Sentences		
Relation	P+	P-	+Cnt	-Cnt	*Cnt
attendedSchool	83	**97**	541	212	**544**
hasChild	91	**96**	**661**	68	106
hasSpouse	87	**99**	**1262**	157	282
hasSibling	87	**97**	**313**	72	272
GPEEmploys	**70**	60	**1208**	308	313
GPELeader	**73**	69	**1018**	629	644
ORGEmploys	61	**96**	**1698**	142	209
ORGLeader	**92**	82	**1095**	207	286
hasBirthDate	88	**97**	**231**	131	182
hasBirthPlace	**90**	85	**836**	388	558

Table 3: Number of sentences in which each system found relation instances

Our experiments suggest that in contexts where recall is important incorporating coreference into a relation extraction system may provide significant gains. Despite being noisy, coreference information improved F-scores for all relations in our test, more than doubling the F-score for 5 of the 10.

Why is the high error rate of coreference not very harmful to +Coref? We speculate that there are two reasons. First, during training, not all coreference is treated equally. If the only evidence we have for a proposed instance depends on low confidence coreference links, it is very unlikely to be added to our instance set for use in future iterations. Second, for both training and runtime, many of the coreference links relevant for extracting the relation set examined here are fairly reliable, such as *wh*-words in relative clauses.

There is room for more investigation of the question, however. It is also unclear if the same result would hold for a very different set of relations, especially those which are more event-like than relation-like.

Acknowledgments

This work was supported, in part, by DARPA under AFRL Contract FA8750-09-C-179. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We would like to thank our reviewers for their helpful comments and Martha Friedman, Michael Heller, Elizabeth Roman, and Lorna Sigourney for doing our evaluation annotation.