Coreference resolution evaluation for higher level applications

Tuggener, Don

Coreference Resolution Evaluation for Higher Level Applications

Don Tuggener
Unversity of Zurich
Institute of Computational Linguistics
tuggener@cl.uzh.ch

Abstract

This paper presents an evaluation framework for coreference resolution geared towards interpretability for higher-level applications. Three application scenarios for coreference resolution are outlined and metrics for them are devised. The metrics provide detailed system analysis and aim at measuring the potential benefit of using coreference systems in preprocessing.

1 Introduction

Coreference Resolution is often described as an important preprocessing step for higher-level applications. However, the commonly used coreference evaluation metrics (MUC, BCUB, CEAF, BLANC) treat coreference as a generic clustering problem and perform cluster similarity measures to evaluate coreference system outputs. Mentions are seen as unsorted generic items rather than linearly ordered linguistic objects (Chen and Ng, 2013). This makes it arguably hard to interpret the scores and assess the potential benefit of using a coreference system as a preprocessing step.

Therefore, this paper proposes an evaluation framework for coreference systems which aims at bridging the gap between coreference system development, evaluation, and higher level applications. For this purpose, we outline three types of application scenarios which coreference resolution can benefit and devise metrics for them which are easy to interpret and provide detailed system output analysis based on any available mention feature.

2 Basic Concepts

Like other coreference metrics, we adapt the concepts of Recall and Precision from evaluation in Information Retrieval (IR) to compare mentions in a system output (the response) to the annotated mentions in a gold standard (the key). To stay close to the originally clear definitions of Recall and Precision in IR, Recall is aimed at identifying *how many of the annotated key mentions are correctly resolved* by a system, and Precision will measure *the correctness of the returned system mentions*.

However, if we define Recall as $\frac{tp}{tp+fn}$, the denominator will not include key mentions that have been put in the wrong coreference chain, and will not denote *all mentions in the key*. Therefore, borrowing nomenclature from (Durrett and Klein, 2013), we introduce an additional error class, *wrong linkage (wl)*, which signifies key mentions that have been linked to incorrect antecedents. Recall can then be defined as $\frac{tp}{tp+\mathbf{wl}+fn}$ and Precision as $\frac{tp}{tp+\mathbf{wl}+fp}$. Recall then extends over all key mentions, and Precision calculation includes all system mentions.

Furthermore, including *wrong linkage* in the Recall equation prevents it from inflating compared to Precision when a large number of key mentions are incorrectly resolved. Evaluation is also sensitive to the anaphoricity detection problem. For example, an incorrectly resolved *anaphoric* "it" pronoun is counted as *wrong linkage* and thus also affects Recall, while a resolved *pleonastic* "it" pronoun is considered a *false positive* which is only penalized by Precision. Beside the "it" pronoun, this is of particular relevance for noun markables, as determining their referential status is a non-trivial subtask in coreference resolution.

As we evaluate each mention individually, we are able to measure performance regarding any feature type of a mention, e.g. PoS, number, gender, semantic class etc. We will focus on mention types based on PoS tags (i.e. pronouns, nouns etc.), as they are often the building blocks of coreference systems. Furthermore, mention type based

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 231–235,
Gothenburg, Sweden, April 26-30 2014.

performance analysis is informative for higher-level applications, as they might be specifically interested in certain mention types.

3 Application Scenarios

Next, we will outline three higher-level application types which consume coreference and devise relevant metrics for them.

3.1 Models of entity distributions

The first application scenario subsumes models that investigate distributions and patterns of entity occurrences in discourse. For example, Centering theory (Grosz et al., 1995) and the thereof derived entity grid model (Barzilay and Lapata, 2008; Elsner and Charniak, 2011) record transitions of grammatical functions that entities occur with in coherent discourse. These models can benefit from coreference resolution if entities are pronominalized or occur as a non-string matching nominal mentions.

Another application which tracks sequences of entity occurrences is event sequence modeling. Such models investigate prototypical sequences of events to derive event schemes or templates of successive events (Lee et al., 2012; Irwin et al., 2011; Kuo and Chen, 2007). Here, coreference resolution can help link pronominalized arguments of events to their previous mention and, thereby, maintain the event argument sequence.

The outlined applications in this scenario primarily rely on the identification of *correct and gapless sequences of entity occurrences*. We can approximate this requirement in a metric by requiring the immediate antecedent of a mention in a response chain to be the immediate antecedent of that mention in the key chain.

Note that this restriction deems mentions as incorrect, if they skip an antecedent but are resolved to another antecedent in the correct chain. For example, given a key [A-B-C-D], mention D in a response [A-B-D] would not be considered correct, as the immediate antecedent is not the same as in the key. The original sequence of the entity's occurrence is broken between mention B and D in the response, as mention C is missing.

We use the following algorithm (table 1) to calculate Recall and Precision for evaluating immediate antecedents. Let K be the key and S be the system response. Let e be an entity denoted by m_n mentions.

01 for $e_k \in K$:
02 for $m_i \in e_k \wedge i > 0$:
03 if $\neg\exists e_s, m_j : (e_s \in S \wedge m_j \in e_s \wedge m_j = m_i \wedge \exists predecessor(m_j, e_s)) \rightarrow$ fn++
04 elif $\exists e_s, m_j : (e_s \in S \wedge m_j \in e_s \wedge m_j = m_i \wedge predecessor(m_i, e_k) = predecessor(m_j, e_s)) \rightarrow$ tp++
05 else wl++
06 for $e_s \in S$:
07 for $m_i \in e_s \wedge i > 0$:
08 if $\neg\exists e_k, m_j : (e_k \in K \wedge m_j \in e_k \wedge m_j = m_i \wedge \exists predecessor(m_j, e_k)) \rightarrow$ fp++

Table 1: Algorithm for calculating Recall and Precision.

We traverse the key K and each entity e_k in it[1]. We evaluate each mention m_i in e_k, except for the first one (line 2), as we investigate coreference links. If no response chain exists that contains m_i and its predecessor, we count m_i as a false negative (line 3). This condition subsumes the case where m_i is not in the response, and the case where m_i is the first mention of a response chain. In the latter case, the system has deemed m_i to be non-anaphoric (i.e. the starter of a chain), while it is anaphoric in the key[2]. We check whether the immediate predecessor of m_i in the key chain e_k is also the immediate predecessor of m_j in the response chain e_s (line 4). If true, we count m_i as a true positive, or as wrong linkage otherwise.

We traverse the response chains to detect spurious system mentions, i.e. mentions not in the key, and count them as false positives, i.e. non-anaphoric markables that have been resolved by the system (lines 6-8). Here, we also count mentions in the response, which have no predecessor in a key chain, as false positives. If a mention in the response chain is the chain starter in a key chain, it means that the system has falsely deemed it to be anaphoric and we regard it as a false positive[3].

3.2 Inferred local entities

The second application scenario relies on coreference resolution to infer local nominal antecedents. For example, in Summarization, a target sentence may contain a pronoun which should be replaced by a nominal antecedent to avoid ambiguities and ensure coherence in the summary. Machine Trans-

[1] We disregard singleton entities, as it is not clear what benefit a higher level application could gain from them.

[2] (Durrett and Klein, 2013) call this error *false new (FN)*.

[3] This error is called *false anaphoric (FA)* by (Durrett and Klein, 2013).

lation can benefit from pronoun resolution in language pairs where nouns have grammatical gender. In such language pairs, the gender of a pronoun antecedent has to be retrieved in the source language in order to insert the pronoun with the correct gender in the target language.

In these applications, it is not sufficient to link pronouns to other pronouns of the same coreference chain because they do not help infer the underlying entity. Therefore, in our metric, we require the closest preceding nominal antecedent of a mention in a response chain to be an antecedent in the key chain.

The algorithm for calculation of Recall and Precision is similar to the one in table 1. We modify lines 3 and 4 to require the closest nominal antecedent of m_i in the response chain e_s to be an antecedent of m_j in the corresponding key chain e_k, where $m_j = m_i$, i.e.:

$\exists m_h \in e_s : is\_closest\_noun(m_h, m_i) \wedge \exists e_k, m_j, m_l : (e_k \in K \wedge m_j \in e_k \wedge m_j = m_i \wedge m_l \in e_k \wedge l < j \wedge m_l = m_h) \rightarrow$ tp++

Note that we cannot process chains without a nominal mention in this scenario[4]. Therefore, we skip evaluation for such $e_k \in K$. We still want to find incorrectly inferred nominal antecedents of anaphoric mentions, i.e. mentions in $e_s \in S$ that have been assigned a nominal antecedent in the response but have none in the key and count them as wrong linkage, as they infer an incorrect nominal antecedent. Therefore, we traverse all $e_s \in S$ and add to the algorithm:

$\forall m_i \in e_s : \neg is\_noun(m_i) \wedge \exists m_h \in e_s : is\_noun(m_h) \wedge \exists e_k, m_j : (e_k \in K \wedge m_j \in e_k \wedge m_j = m_i \wedge \neg \exists m_l \in e_k : is\_noun(m_l)) \rightarrow$ wl++

3.3 Finding contexts for a specific entity

The last scenario we consider covers applications that are primarily query driven. Such applications search for references to a given entity and analyze or extract its occurrence contexts. For example, Sentiment Analysis searches large text collections for occurrences of a target entity and then derives polarity information from its contexts. Biomedical relation mining looks for interaction contexts of specific genes or proteins etc.

[4]We found that 476 of 4532 key chains (10.05%) do not contain a nominal mention. Furthermore, we do not treat cataphora (i.e. pronouns at chain start) in this scenario. We found that 241 (5.31%) of the key chains start with cataphoric pronouns.

For these applications, references to relevant entities have to be accessible by queries. For example, if a sentiment system investigates polarity contexts of the entity "Barack Obama", given a key chain [Obama - the president - he], a response chain [the president - he] is not sufficient, because the higher level application is not looking for instances of the generic "president" entity.

Therefore, we determine an *anchor mention* for each coreference chain which represents the most likely unique surface form an entity occurs with. As a simple approximation, we choose the first nominal mention of a coreference chain to be the anchor of the entity, because first mentions of entities introduce them to discourse and are, therefore, generally informative, unambiguous, semantically extensive and are likely to contain surface forms a higher level application will query.

Entity Detection
01 for $e_k \in K$:
02 if $\exists m_n \in e_k : is\_noun(m_n)$ $\rightarrow m_{anchor} = determine\_anchor(e_k)$
03 if $\exists m_{anchor} \wedge \exists e_s \in S : m_{anchor} \in e_s \rightarrow$ tp++
04 else $\rightarrow$ fn++
05 for $e_s \in S$:
06 if $\exists m_n \in e_s : is\_noun(m_n)$ $\rightarrow m_{anchor} = determine\_anchor(e_s)$
07 if $\neg \exists e_k \in K : m_{anchor} \in e_k \rightarrow$ fp++
Entity Mentions
01 for $e_k \in K : \exists m_{anchor} \wedge \exists e_s \in S : m_{anchor} \in e_s :$
02 for $m_i \in e_k$:
03 if $m_i \in e_s \rightarrow$ tp++
04 else$\rightarrow$ fn++
05 for $m_i \in e_s$:
06 if $m_i \neg \in e_k \rightarrow$ fp++

Table 2: Algorithm for calculating Recall and Precision using anchor mentions.

To calculate Recall and Precision, we align coreference chains in the responses to those in the key via their anchors and then measure how many (in)correct references to that anchor the coreference systems find (table 2). We divide evaluation into *entity detection* (ED), which measures how many of the anchor mentions a system identifies. We then measure the quality of the *entity mentions* (EM) for only those entities which have been aligned through their anchors.

The quality of the references to the anchor mentions are not directly comparable between systems, as their basis is not the same if the number of aligned anchors differs. Therefore, we calculate the harmonic mean of entity detection and entity mentions to enable direct system compari-

son. Where applicable, we obtain the named entity class of the entity and measure performance for each such class.

4 Evaluation

We apply our metrics to three available coreference systems, namely the Berkley system (Durrett and Klein, 2013), the IMS system (Björkelund and Farkas, 2012), and the Stanford system (Lee et al., 2013) and their responses for the CoNLL 2012 shared task test set for English (Pradhan et al., 2012). Tables 3 and 4 report the results.

	Immediate antecedent			Inferred antecedent		
	R	P	F	R	P	F
	BERK (Durrett and Klein, 2013)					
NOUN	45.06	47.06	46.04	55.54	60.37	57.85
PRP	67.66	64.87	66.24	48.92	53.62	51.16
PRP$	74.49	74.32	74.41	61.95	66.80	64.28
TOTAL	56.60	56.91	56.76	52.94	58.04	55.37
	IMS (Björkelund and Farkas, 2012)					
NOUN	38.01	43.09	40.39	46.90	54.96	50.61
PRP	**69.06**	**68.64**	**68.85**	43.04	**57.42**	49.20
PRP$	72.57	72.11	72.34	51.51	63.54	56.90
TOTAL	53.55	57.55	55.48	45.27	56.47	50.25
	STAN (Lee et al., 2013)					
NOUN	38.51	42.92	40.60	50.03	57.62	53.56
PRP	65.55	61.09	63.25	36.67	45.97	40.80
PRP$	66.12	65.70	65.91	40.64	52.38	45.77
TOTAL	51.70	52.69	52.19	43.01	51.73	46.97

Table 3: Antecedent based evaluation

We note that the system ranking based on the MELA score[5] is retained by our metrics. MELA rates the Berkley system best (61.62), followed by the IMS system (57.42), and then the Stanford system (55.69).

Beside detailed analysis based on PoS tags, our metrics reveal interesting nuances. Somewhat expectedly, noun resolution is worse when the immediate antecedent is evaluated, than if the next nominal antecedent is analyzed. Symmetrically inverse, pronouns achieve higher scores when their direct antecedent is measured, as compared to when the next nominal antecedent has to be correct.

Our evaluation shows that the IMS system achieves a higher score for pronouns than the Berkley system when immediate antecedents are measured and has a higher Precision for pronouns regarding the inferred antecedents. The Berkley system performs best mainly due to Recall. For e.g. personal pronouns (PRP), Berkley has the following counts for the inferred antecedents: tp=2687, wl=1935, **fn=871**, fp=389, while IMS shows tp=2243, wl=1376, **fn=1592**, fp=287. This indicates that the IMS Recall is lower because of the high false negative count, rather than being due to too many wrong linkages.

Finally, table 4 suggests that the IMS systems performs significantly worse in the PERSON class than the other systems and is outperformed by the Stanford system in the ORG class, but performs best in the GPE class.

		R	P	F	Fϕ
		PERSON (18.69%)			
BERK	ED	64.02	75.88	69.45	67.11
	EM	63.60	66.29	64.92	
IMS	ED	45.66	51.69	48.48	**52.74**
	EM	47.67	73.45	57.82	
STAN	ED	56.33	59.74	57.98	61.61
	EM	53.84	84.37	65.73	
		GPE (13.28%)			
BERK	ED	73.21	77.36	75.23	75.71
	EM	69.89	83.73	76.19	
IMS	ED	73.51	74.17	73.84	**76.21**
	EM	69.94	90.04	78.73	
STAN	ED	70.24	76.62	73.29	75.24
	EM	68.44	88.81	77.30	
		ORG (9.63%)			
BERK	ED	62.78	67.13	64.88	67.62
	EM	66.87	74.78	70.60	
IMS	ED	44.98	54.30	49.20	56.85
	EM	57.26	81.66	67.32	
STAN	ED	49.68	58.56	53.75	**59.41**
	EM	57.25	79.05	66.41	
		TOTAL (100%)			
BERK	ED	58.65	53.19	55.79	63.41
	EM	72.65	74.28	73.45	
IMS	ED	47.16	42.66	44.80	55.24
	EM	65.88	79.40	72.01	
STAN	ED	48.62	41.40	44.72	55.27
	EM	65.66	80.48	72.32	

Table 4: Anchor mention based evaluation

5 Conclusion

We have presented a simple evaluation framework for coreference evaluation with higher level applications in mind. The metrics allow specific performance measurement regarding different antecedent requirements and any mention feature, such as PoS type, lemma, or named entity class, which can aid system development and comparison. Furthermore, the metrics do not alter system rankings compared to the commonly used evaluation approach[6].

[5] $\frac{MUC+BCUB+CEAFE}{3}$

[6] The scorers are freely available on our website: http://www.cl.uzh.ch/research/coreferenceresolution.html

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Comput. Linguist.*, 34(1):1–34, March.

Anders Björkelund and Richárd Farkas. 2012. Data-driven multilingual coreference resolution using resolver stacking. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 49–55, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen Chen and Vincent Ng. 2013. Linguistically aware coreference evaluation metrics. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 1366–1374.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, October. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2011. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 125–129, Stroudsburg, PA, USA. Association for Computational Linguistics.

Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: a framework for modeling the local coherence of discourse. *Comput. Linguist.*, 21(2):203–225, June.

Joseph Irwin, Mamoru Komachi, and Yuji Matsumoto. 2011. Narrative schema as world knowledge for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 86–92, Stroudsburg, PA, USA. Association for Computational Linguistics.

June-Jei Kuo and Hsin-Hsi Chen. 2007. Cross-document event clustering using knowledge mining from co-reference chains. *Inf. Process. Manage.*, 43(2):327–343, March.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA. Association for Computational Linguistics.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4).

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.