A Joint Model for Quotation Attribution and Coreference Resolution

Mariana S. C. Almeida∗† Miguel B. Almeida∗† André F. T. Martins∗†

∗Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal
†Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
{mla,mba,atm}@priberam.pt

Abstract

We address the problem of automatically attributing quotations to speakers, which has great relevance in text mining and media monitoring applications. While current systems report high accuracies for this task, they either work at mention-level (getting credit for detecting uninformative mentions such as pronouns), or assume the coreferent mentions have been detected beforehand; the inaccuracies in this preprocessing step may lead to error propagation. In this paper, we introduce a joint model for entity-level quotation attribution and coreference resolution, exploiting correlations between the two tasks. We design an evaluation metric for attribution that captures all speakers’ mentions. We present results showing that both tasks benefit from being treated jointly.

1 Introduction

Quotations are a crucial part of news stories, giving the perspectives of the participants in the narrated event, and making the news sound objective. The ability of extracting and organizing these quotations is highly relevant for text mining applications, as it may aid journalists in fact-checking, help users browse news threads, and reduce human intervention in media monitoring. This involves assigning the correct speaker to each quote—a problem called quotation attribution (§2).

There is significant literature devoted to this task, both for narrative genres (Mamede and Chaleira, 2004; Elson and McKeown, 2010) and newswire domains (Pouliquen et al., 2007; Sarmento et al., 2009; Schneider et al., 2010). While the earliest works focused on devising lexical and syntactic rules and hand-crafting grammars, there has been a recent shift toward machine learning approaches (Fernandes et al., 2011; O’Keefe et al., 2012; Pareti et al., 2013), with latest works reporting high accuracies for speaker identification in newswire (in the range 80–95% for direct and mixed quotes, according to O’Keefe et al. (2012)). Despite these encouraging results, quotation mining systems are not yet fully satisfactory, even when only direct quotes are considered. Part of the problem, as we next describe, has to do with inaccuracies in coreference resolution (§3).

The “easiest” instances of quotation attribution problems arise when the speaker and the quote are semantically connected, e.g., through a reported speech verb like said. However, in newswire text, the subject of this verb is commonly a pronoun or another uninformative anaphoric mention. While the speaker thus determined may well be correct—being in most cases consistent with human annotation choices (Pareti, 2012)—from a practical perspective, it will be of little use without a coreference system that correctly resolves the anaphora. Since the current state of the art in coreference resolution is far from perfect, errors at this stage tend to propagate to the quote attribution system.

Consider the following examples for illustration (taken from the WSJ-1057 and WSJ-0089 documents in the Penn Treebank), where we have annotated with subscripts some of the mentions:

(a) Rivals carp at “the principle of [Pilson]M1,” as [NBC’s Arthur Watson]M2 once put it—“[he]M3’s always expounding that rights are too high, then [he]M4’s going crazy.” But [the 49-year-old Mr. Pilson]M5 is hardly a man to ignore the numbers.


In example (a), the pronoun coreference system used by O’Keefe et al. (2012) erroneously clusters together mentions M2, M3 and M4 (instead of the correct clustering {M1, M3, M4}). Since it is unlikely that the speaker is co-referent to a third-
person pronoun *he* inside the quote, a pipeline system would likely attribute (incorrectly) this quote to *Pilson*. In example (b), there are two quotes with the same speaker entity (as indicated by the cue *she added*). This gives evidence that $M_1$ and $M_6$ should be coreferent. A pipeline approach would not be able to exploit these correlations.

We argue that this type of mistakes, among others, can be prevented by a system that performs quote attribution and coreference resolution jointly (§4). Our joint model is inspired by recent work in coreference resolution that independently ranks the possible mention’s antecedents, forming a latent coreference tree structure (Denis and Baldridge, 2008; Fernandes et al., 2012; Durrett et al., 2013; Durrett and Klein, 2013). We consider a generalization of these structures which we call a quotation-coreference tree. To effectively couple the two tasks, we need to go beyond simple arc-factored models and consider paths in the tree. We formulate the resulting problem as a logic program, which we tackle using a dual decomposition strategy (§5). We provide an empirical comparison between our method and baselines for each of the tasks and a pipeline system, defining suitable metrics for entity-level quotation attribution (§6).

## 2 Quotation Attribution

The task of quotation attribution can be formally defined as follows. Given a document containing a sequence of quotations, $\langle q_1, \ldots, q_L \rangle$, and a set of candidate speakers, $\{s_1, \ldots, s_M\}$, the goal is to assign a speaker to every quote.

Previous work has handled direct and mixed quotations (Sarmento et al., 2009; O’Keefe et al., 2012), easily extractable with regular expressions for detecting quotation marks, as well as indirect quotations (Pareti et al., 2013), which are more involved and require syntactic or semantic patterns. In this work, we resort to direct and mixed quotations. Pareti (2012) defines quotation attributions in terms of their content span (the quotation text itself), their cue (a lexical anchor of the attribution relation, such as a reported speech verb), and the source span (the author of the quote). The same reference introduced the PARC dataset, which we use in our experiments (§6) and which is based on the annotation of a database of attribution relations from the Penn Discourse Treebank (Prasad et al., 2008). Several machine learning algorithms have been applied to this task, either framing the problem as classification (an independent decision for each quote), or sequence labeling (using greedy methods or linear-chain conditional random fields); see O’Keefe et al. (2012) for a comparison among these different methods.

In this paper, we distinguish between mention-level quotation attribution, in which the candidate speakers are individual mentions, and entity-level quotation attribution, in which they are entity clusters comprised of one or more mentions. With this distinction, we attempt to clarify how prior work has addressed this task, and design suitable baselines and evaluation metrics. For example, O’Keefe et al. (2012) applies a coreference resolver before quotation attribution, whereas de La Clergerie et al. (2011) does it afterwards, as a post-processing stage. An important issue when evaluating quotation attribution systems is to prevent them from getting credit for detecting uninformative speakers such as pronouns; we will get back to this topic in §6.2.

## 3 Coreference Resolution

In coreference resolution, we are given a set of mentions $M := \{m_1, \ldots, m_K\}$, and the goal is to cluster them into discourse entities, $E := \{e_1, \ldots, e_J\}$, where each $e_j \subseteq M$ and $e_j \neq \emptyset$. We follow Haghighi and Klein (2007) and distinguish between proper, nominal, and pronominal mentions. Each requires different types of information to be resolved. Thus, the task involves determining anaphoricity, resolving pronouns, and identifying semantic compatibility among mentions. To resolve these references, one typically exploits contextual and grammatical clues, as well as semantic information and world knowledge, to understand whether mentions refer to people, places, organizations, and so on. The importance of coreference resolution has led to it being the subject of recent CoNLL shared tasks (Pradhan et al., 2011; Pradhan et al., 2012).

There has been a variety of approaches for this problem. Early work used local discriminative classifiers, making independent decisions for each mention or pair of mentions (Soon et al., 2001; Ng and Cardie, 2002). Lee et al. (2011) proposed a competitive non-learned sieve-based method, which constructs clusters by agglomerating mentions in a greedy manner. Entity-centric models define scores for the entire entity clusters (Culotta et al., 2007; Haghighi and Klein, 2010;
Rahman and Ng, 2011) and seek the set of entities that optimize the sum of scores; this can also be promoted in a decentralized manner (Durrett et al., 2013). Pairwise models (Bengtson and Roth, 2008; Finkel et al., 2008; Versley et al., 2008), on the other hand, define scores for each pair of mentions to be coreferent, and define the clusters as the transitive closure of these pairwise relations. A disadvantage of these two methods is that they lead to intractable decoding problems, so approximate methods must be used. For comprehensive overviews, see Stoyanov et al. (2009), Ng (2010), Pradhan et al. (2011) and Pradhan et al. (2012).

Our joint approach (to be fully described in §4) draws inspiration from recent work that shifts from entity clusters to coreference trees (Fernandes et al., 2012; Durrett and Klein, 2013). These models define scores for each mention to link to its antecedent or to an artificial root symbol $\$$(in which case it is not anaphoric). The computation of the best tree can be done exactly with spanning tree algorithms, or by independently choosing the best antecedent (or the root) for each mention, if only left-to-right arcs are allowed. The same idea underlies the antecedent ranking approach of Denis and Baldridge (2008). Once the coreference tree is computed, the set of entity clusters $E$ is obtained by associating each entity set to a branch of the tree coming out from the root. This is illustrated in Figure 1 (left).

4 Joint Quotations and Coreferences

In this work, we propose that quotation attribution and coreference resolution are solved jointly by treating both mentions and quotations as nodes in a generalized structure called a quotation-coreference tree (Figure 1, right). The joint system’s decoding process consists in creating such a tree, from which a clustering of the nodes can be immediately obtained. The clustering is interpreted as follows:

- All mention nodes in the cluster are coreferent, thus they describe one single entity (just like in a standard coreference tree).
- Quotation nodes that appear together with those mentions in a cluster will be assigned that entity as the speaker.

For example, in Figure 1 (right), the entity Dorothy L. Sayers (formed by mentions $\{M_1, M_6\}$) is assigned as the speaker of quotations $Q_1$ and $Q_2$. We forbid arcs between quotes and from a quote to a mention, effectively constraining the quotes to be leaves in the tree, with mentions as parents.\footnote{This is implemented by defining $-\infty$ scores for all the outgoing arcs in a quotation node, as well as incoming arcs originating from the root.} We force a tree with only left-to-right arcs, by choosing a total ordering of the nodes that places all the quotations in the rightmost positions (which implies that any arc connecting a mention to a quotation will point to the right). The quotation-coreference tree is obtained as the best spanning tree that maximizes a score function, to be described next.

4.1 Basic Model

Our basic model is a feature-based linear model which assigns a score to each candidate arc linking two mentions (mention-mention arcs), or linking a mention to a quote (mention-quotation arcs). Our basic system is called QUOTEBEFORECOREF for reasons we will detail in section 4.2.

4.1.1 Coreference features

For the mention-mention arcs, we use the same coreference features as the SURFACE model of the Berkeley Coreference Resolution System (Durrett and Klein, 2013), plus features for gender and number obtained through the dataset of Bergsma and Lin (2006). This is a very simple lexical-driven model which achieves state-of-the-art results. The features are shown in Table 1.

4.1.2 Quotation features

For the quote attribution features, we use features inspired by O’Keefe et al. (2012), shown in Table 2. The same set of features works for speakers that are individual mentions (in the model just described), and for speakers that are clusters of mentions (used in §6 for the baseline QUOTEAFTERCOREF). These features include various distances between the mention and the quote, the indication of the speaker being inside the quote span, and various contextual features.

4.2 Final Model

While the basic model just described puts quotations and mentions together, it is not more expressive than having separate models for the two tasks. In fact, if we just have scores for individual arcs, the two problems are decoupled: the optimal
Figure 1: Left: A typical coreference tree for the text snippet in §1, example (b), with mentions $M_1$ and $M_6$ clustered together and $M_2$ and $M_3$ left as singletons. Right: A quotation-coreference tree for the same example. Mention nodes are depicted as green circles, and quotation nodes in shaded blue. The dashed rectangle represents a branch of the tree, containing the entity cluster associated with the speaker Dorothy L. Sayers, as well as the quotes she authored.

<table>
<thead>
<tr>
<th>Features on the child mention</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ANAPHORIC (T/F)] + [CHILD HEAD WORD]</td>
</tr>
<tr>
<td>[ANAPHORIC (T/F)] + [CHILD FIRST WORD]</td>
</tr>
<tr>
<td>[ANAPHORIC (T/F)] + [CHILD LAST WORD]</td>
</tr>
<tr>
<td>[ANAPHORIC (T/F)] + [CHILD PRECEDING WORD]</td>
</tr>
<tr>
<td>[ANAPHORIC (T/F)] + [CHILD FOLLOWING WORD]</td>
</tr>
<tr>
<td>[ANAPHORIC (T/F)] + [CHILD LENGTH]</td>
</tr>
<tr>
<td>Features on the parent mention</td>
</tr>
<tr>
<td>[PARENT HEAD WORD]</td>
</tr>
<tr>
<td>[PARENT FIRST WORD]</td>
</tr>
<tr>
<td>[PARENT LAST WORD]</td>
</tr>
<tr>
<td>[PARENT PRECEDING WORD]</td>
</tr>
<tr>
<td>[PARENT FOLLOWING WORD]</td>
</tr>
<tr>
<td>[PARENT LENGTH]</td>
</tr>
<tr>
<td>[PARENT GENDER]</td>
</tr>
<tr>
<td>[PARENT NUMBER]</td>
</tr>
<tr>
<td>Features on the pair</td>
</tr>
<tr>
<td>[EXACT STRING MATCH (T/F)]</td>
</tr>
<tr>
<td>[HEAD MATCH (T/F)]</td>
</tr>
<tr>
<td>[SENTENCE DISTANCE, CAPPED AT 10]</td>
</tr>
<tr>
<td>[MENTION DISTANCE, CAPPED AT 10]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features on the quote-speaker pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>[WORD DISTANCE]</td>
</tr>
<tr>
<td>[SENTENCE DISTANCE]</td>
</tr>
<tr>
<td>[# IN-BETWEEN QUOTES]</td>
</tr>
<tr>
<td>[# IN-BETWEEN SPEAKERS]</td>
</tr>
<tr>
<td>[SPEAKER IN QUOTE, 1ST PERS. SG. PRONOUN (T/F)]</td>
</tr>
<tr>
<td>[SPEAKER IN QUOTE, 1ST PERS. PL. PRONOUN (T/F)]</td>
</tr>
<tr>
<td>[SPEAKER IN QUOTE, OTHER (T/F)]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features on the speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PREVIOUS WORD IS QUOTE (T/F)]</td>
</tr>
<tr>
<td>[PREVIOUS WORD IS SAME QUOTE (T/F)]</td>
</tr>
<tr>
<td>[PREVIOUS WORD IS ANOTHER QUOTE (T/F)]</td>
</tr>
<tr>
<td>[PREVIOUS WORD IS SPEAKER (T/F)]</td>
</tr>
<tr>
<td>[PREVIOUS WORD IS PUNCTUATION (T/F)]</td>
</tr>
<tr>
<td>[PREVIOUS WORD IS REPORTED SPEECH VERB (T/F)]</td>
</tr>
<tr>
<td>[PREVIOUS WORD IS VERB (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS QUOTE (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS SAME QUOTE (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS ANOTHER QUOTE (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS SPEAKER (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS PUNCTUATION (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS REPORTED SPEECH VERB (T/F)]</td>
</tr>
<tr>
<td>[NEXT WORD IS VERB (T/F)]</td>
</tr>
</tbody>
</table>

Table 1: Coreference features, associated to each candidate mention-mention arc in the tree. As in Durrett and Klein (2013), we also include conjunctions of each feature with the child and parent mention types (proper, nominal, or, if pronominal, the pronoun word).

Table 2: Quotation attribution features, associated to each quote-speaker candidate. These features are used in the QUOTEONLY, QUOTE-BEFORECOREF, and JOIN systems (where the speaker is a mention) and in the QUOTEAFTERCOREF system (where the speaker is an entity).

quotation-coreference tree can be obtained by first assigning the highest scored mention to each quotation, and then building a standard coreference tree involving only the mention nodes. This corresponds to the QUOTE-BEFORECOREF baseline, to be used in §6.

To go beyond separate models, we introduce a final JOIN model, which includes additional scores that depend not just on arcs, but also on paths in the tree. Concretely, we select certain pairs of nodes and introduce scores for the event that both nodes are in the same branch of the tree. Rather than doing this for all pairs—which essentially would revert to the computationally demanding pairwise coreference models discussed in §3—we focus on a small set of pairs that are mostly related with the interaction between the two tasks we address jointly. Namely, we consider the mention-quotation pairs such that the mention
Table 3: Features used in the JOINT system for mention-quote pairs (only for mentions inside quotes) and for quote pairs (only for consecutive quotes). These features are associated to pairs in the same branch of the quotation-coreference tree.

span is within the quotation span (mention-inside-quotations pairs), and pairs of quotations that appear consecutively in the document (consecutive quotation pairs). The idea is that, if consecutive quotations appear on the same branch of the tree, they will have the same speaker (the entity class associated with that branch), even though they are not necessarily siblings. These two pairs are aligned with the motivating examples (a) and (b) shown in §1.

### 4.2.1 Mention-inside-quotations features

The top rows of Table 3 show the features we defined for mentions inside quotes. The features indicate whether the mention is first-person singular pronoun (I, me, my, myself), which provides strong evidence that it co-refers with the quotation author, whether it is first-person plural pronoun (we, us, our, ourselves), which provides a weaker evidence (but sometimes works for collective entities that are organizations), and whether none of the above happens—in which case, the speaker is unlikely to be co-referent with the mention.

### 4.2.2 Consecutive quotation features

We show our consecutive quote features in the bottom rows of Table 3. We use only distance features, measuring both distance in sentences and in words, with binning. These simple features are enough to capture the trend of consecutive quotes that are close apart to have the same speaker.

### 5 Joint Decoding and Training

While decoding in the basic model is easy—as pointed out above, it can even be done by running a mention-level quotation attributor and the coreference resolver independently (QUOTEBEFORECORE)—exact decoding with the JOINT model is in general intractable, since this model breaks the independence assumption between the arcs. However, given the relatively small amount of node pairs that have scores (only mentions inside quotations and consecutive quotations), we expect this “perturbation” to be small enough not to affect the quality of an approximate decoder. The situation resembles other problems in NLP, such as non-projective dependency parsing, which becomes intractable if higher order interactions between the arcs are considered, but can still be well approximated. Inspired by work in parsing (Martins et al., 2009) using linear relaxations with multi-commodity flow models, we propose a similar strategy by defining auxiliary variables and coupling them in a logic program.

### 5.1 Logic Formulation

We next derive the logic program for joint decoding of coreferences and quotations. The input is a set of nodes (including an artificial node), a set of candidate arcs with scores, and a set of node pairs with scores. To make the exposition lighter, we index nodes by integers (starting by the root node 0) and we do not distinguish between mention and quotation nodes. Only arcs from left to right are allowed. The variables in our logic program are:

- **Arc variables** $a_{i \rightarrow j}$, which take the value 1 if there is an arc from $i$ to $j$, and 0 otherwise.
- **Pair variables** $p_{i,j}$, which indicate that nodes $i$ and $j$ are in the same branch of the tree.
- **Path variables** $\pi_{j \rightarrow *k}$, indicating if there is a path from $j$ to $k$.
- **Common ancestor variables** $\psi_{i \rightarrow *jk}$, indicating that node $i$ is a common ancestor of nodes $j$ and $k$ in the tree.

Consistency among these variables is ensured by the following set of constraints:

- Each node except the root has exactly one parent:
  \[
  \sum_{i=0}^{j-1} a_{i \rightarrow j} = 1, \quad \forall j \neq 0 \tag{1}
  \]
- There is a path from each node to itself:
  \[
  \pi_{i \rightarrow *i} = 1, \quad \forall i \tag{2}
  \]
- There is a path from $i$ to $k$ iff there is some $j$ such that $i$ is connected to $j$ and there is path...
from $j$ to $k$:

$$\pi_{i \rightarrow k} = \bigvee_{i < j \leq k} (a_{i \rightarrow j} \land \pi_{j \rightarrow k}), \forall i, k$$  \hspace{1cm} (3)

- Node $i$ is a common ancestor of $k$ and $\ell$ iff there is a path from $i$ to $k$ and from $i$ to $\ell$:

$$\psi_{i \rightarrow k, \ell} = \pi_{i \rightarrow k} \land \pi_{i \rightarrow \ell}, \forall i, k, \ell$$  \hspace{1cm} (4)

- Nodes $k$ and $\ell$ are in the same branch if they have a common ancestor which is not the root:

$$p_{k, \ell} = \bigvee_{i \neq 0} \psi_{i \rightarrow k, \ell}, \forall k, \ell$$  \hspace{1cm} (5)

The objective to optimize is linear in the arc and pair variables (hence the problem can be represented as an integer linear program by turning the logical constraints into linear inequalities).

5.2 Dual Decomposition

To decode, we employ the alternating directions dual decomposition algorithm (AD$^3$), which solves a relaxation of the ILP above. AD$^3$ has been used successfully in various NLP tasks, such as dependency parsing (Martins et al., 2011; Martins et al., 2013), semantic role labeling (Das et al., 2012), and compressive summarization (Almeida and Martins, 2013). At test time, if the solution is not integer, we apply a simple rounding procedure to obtain an actual tree: for each node $j$, obtain the antecedent (or root) $i$ with the highest $a_{i \rightarrow j}$, solving ties arbitrarily.

5.3 Learning the Model

We train the joint model with the max-loss variant of the MIRA algorithm (Crammer et al., 2006), adapted to latent variables (we simply obtain the best tree consistent with the gold clustering at each step of MIRA, before doing cost-augmented decoding). The resulting algorithm is very similar to the latent perceptron algorithm in Fernandes et al. (2011), but it uses the aggressive stepsize of MIRA. We set the same costs for coreference mistakes as Durrett and Klein (2013), and a unit cost for missing the correct speaker of a quotation. For speeding up decoding, we first train a basic pruner for the coreference system (using only the features described in §4.1.1), limiting the number of candidate antecedents to 10, and discarding scores whose difference with respect to the best antecedent is below a threshold. We also freeze the best coreference trees consistent with the gold clustering using the pruner model, to eliminate the need of latent variables in the second stage.

6 Experiments

6.1 Dataset

We used the 597 documents of the Wall Street Journal (WSJ) corpus that were disclosed for the CoNLL-2011 coreference shared task (Pradhan et al., 2011) as a dataset for coreference resolution. This dataset includes train, development and test partitions, annotated with coreference information, as well as gold and automatically generated syntactic and semantic information.

The CoNLL-2011 corpus does not contain annotations of quotation attribution. For that reason, we used the WSJ quotation annotations in the PARC dataset (Pareti, 2012). We used the same version of the corpus as O’Keefe et al. (2012), but with different splits, to match the dataset partitions in the coreference resolution data. This attribution corpus contains 279 documents of the 597 CoNLL-2011 files, having a total of 1199 annotated quotes. As in that work, we only considered directed speech quotes and the direct part of mixed quotes (quotes with both direct and undirected speech).

6.2 Metrics for quotation attribution

Previous evaluations of quotation attribution systems were designed at mention level, and are thus assessed by comparing the predicted speaker mention span with the gold one. This metric assesses the amount of speaker mentions that were correctly identified. For compatibility with previous assessments, we report this score, which we call Exact Match (EM): this is the percentage of predicted speakers with the same span as the gold one.

However, for several quotations (about 30% in the PARC corpus) this information is of little value, since the gold mention is a pronoun, which per se does not give any useful information about the actual speaker entity. Considering this fact, we propose two other metrics that capture information at the entity level, reflecting the amount of information a system is able to extract about the speakers:

- Representative Speaker Match (RSM): for each annotated quote, we obtain the full gold coreference set of the gold annotated speaker, and
choose a representative speaker from that cluster. We define this representative speaker as the proper mention which is the closest to the quote (if available); if the cluster does not contain proper mentions, we use the closest nominal mention; if only pronominal mentions are available, we use the original annotated speaker. The final measure is the percentage of predicted speakers that match the string of the corresponding representative speakers.

- **Entity Cluster F₁ (ECF₁).** Considering that a system outputs a set of mentions coreferent to the predicted speakers, we compute the F₁ score between the predicted set and the gold coreference cluster of the correct speaker.

The entity level metrics are not only useful for assessing the quality of an quotation attribution system—they also reflect the quality of the underlying coreference system used to cluster the related mentions.

### 6.3 Attribution baselines

To analyze the task of entity-level quotation attribution, we implemented three baseline systems.

- **QUOTEONLY:** A quotation attribution system trained on the representative speaker, instead of the gold speaker. For fairness, this baseline was trained with an extra feature indicating the type of the mention (nominal, pronominal or proper).

- **QUOTEAFTERCORE:** An attribution system directly applied to the output of a predicted coreference chain. This baseline uses a coreference pre-processing, as applied in O’Keefe et al. (2012).

- **QUOTEBEFORECORE:** An attribution system trained on the gold speaker, and post-combined with the output of a coreference system. This system should be able to provide a set of informative mentions about a quote, post-resolving the problem of the pronominal speakers. This kind of post-coreference approach was used by de La Clergerie et al. (2011).

### 6.4 Coreference Resolution

We use the coreference results of our basic QUOTEBEFORECORE system as a baseline for coreference resolution. Since this system effectively solves the two problems separately, this can be considered our implementation of the SURFACE system of Durrett and Klein (2013). As reported in Table 4, the performance of our baseline is comparable with the one of the SURFACE system of Durrett and Klein (2013), which is denoted as SURFACE-DK-2013.

Table 4 also show the CoNLL metrics obtained for the proposed system of joint coreference resolution and quotation attribution. Our joint system outperformed the baseline with statistical significance (with \( p < 0.05 \) and according to a bootstrap resampling test (Koehn, 2004)) for all metrics except for the CEAFE F₁ measure, whose value was only slightly improved. These results confirm that the coreference resolution task benefits for being tackled jointly with quotation attribution.

### 6.5 Quotation attribution

We implemented and trained the three attribution systems that were described in §6.3 and the system for joint coreference and author attribution that is detailed in §4. For each system, Table 5 shows the mention-based and entity-based metrics that were described in §6.2.

Training a quotation attribution system using representative speakers instead of the gold speakers (QUOTEONLY) leads to rather disappointing results. As expected, we conclude that assigning the semantically related speaker is considerably easier than selecting another mention that is coreferent with the correct speaker.

Using (predicted) coreference information, both QUOTEAFTERCORE and QUOTEBEFORECORE systems considerably increase our entity-based metrics. This was also expected, since the coreference chain allows these baselines to output a set of related mentions. We observed that, using the coreference resolution clusters as the attribution entity (QUOTEAFTERCORE) influences the results negatively when compared to a more basic system that runs coreference on top of attribution result of the QUOTEONLY system (QUOTEBEFORECORE). These results indicate that the quotation attribution task performs better by looking at the speaker mention that connects more strongly with the quotation, instead of trying to match the whole cluster.

Finally, the scores achieved by our JOINT
Table 4: Coreference obtained with the CoNLL scorer (version 5) in the test partition of the WJS corpus, for the SURFACE system of Durrett and Klein (2013), our baseline implementation of the that system (SURFACE-OURS), and our JOINT approach. All systems were trained in the WSJ portion of the Ontonotes.

<table>
<thead>
<tr>
<th>Model</th>
<th>MUC $F_1$</th>
<th>BCUB $F_1$</th>
<th>CEAFE $F_1$</th>
<th>Avg. $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURFACE-DK-2013</td>
<td>58.87</td>
<td>62.74</td>
<td>45.46</td>
<td>55.7</td>
</tr>
<tr>
<td>SURFACE-OURS [QUOTEBEFORECOREF]</td>
<td>57.89</td>
<td>62.50</td>
<td>45.48</td>
<td>55.3</td>
</tr>
<tr>
<td>JOINT</td>
<td>58.78</td>
<td>63.79</td>
<td>45.50</td>
<td>56.0</td>
</tr>
</tbody>
</table>

Table 5: Attribution results obtained, in the test set, for the three baseline systems and our joint system.

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>RSM</th>
<th>ECF $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUOTEONLY</td>
<td>49.1%</td>
<td>49.4%</td>
<td>41.2%</td>
</tr>
<tr>
<td>QUOTEAFTERCOREF</td>
<td>76.7%</td>
<td>64.6%</td>
<td>70.0%</td>
</tr>
<tr>
<td>QUOTEBEFORECOREF</td>
<td>88.7%</td>
<td>74.7%</td>
<td>73.7%</td>
</tr>
<tr>
<td>JOINT</td>
<td>88.1%</td>
<td>76.6%</td>
<td>74.1%</td>
</tr>
</tbody>
</table>

model are slightly above the best baseline system QUOTEBEFORECOREF, yielding the best performance on the entity-level quotation attribution task. The differences, however, were not found statistically significant, probably due to the small number of quotes (159) in the test set.

The average decoding runtime of the JOINT model is 1.6 sec. per document, against 0.2 sec. for the pipeline system. This slowdown is expectable given the fact that the pipeline system only needs to make independent decisions, while the joint version needs to solve a harder combinatorial problem. Yet, this runtime is within the order of magnitude of the time necessary to preprocess the documents (which includes tagging and parsing the sentences).

6.6 Error Analysis

To understand the type of errors that are prevented with the JOINT system, consider the following example (from document WSJ-2428):

- [Robert Dow, a partner and portfolio manager at Lord, Abbett & Co., $M_1$, which manages $4$ billion of high-yield bonds, says $M_2$ doesn’t “think there is any fundamental economic rationale (for the junk bond rout). It was [herd instinct]$M_3$. “[He]$M_4$ adds: “The junk market has witnessed some trouble and now some people think that if the equity market gets creamed that means the economy will be terrible and that’s bad for junk.”

The basic QUOTEBEFORECOREF system wrongly clusters together $M_3$ and $M_4$ as coreferent, and wrongly assigns $M_3$ as the representative speaker. On the other hand, the JOINT system correctly clusters $M_1$, $M_2$ and $M_4$ as coreferent. This is due to the presence of the consecutive quote features which aid in understanding that both quotes have the same speaker, and the mention-inside-quote features which prevent herd instinct, which is inside a quote, from being coreferent with He, which is very likely the author of the quotes due to the verb adds.

7 Conclusions

We presented a framework for joint coreference resolution and quotation attribution. We represented the problem as finding an optimal spanning tree in a graph including both quotation nodes and mention nodes. To couple the two tasks, we introduce variables that look at paths in the tree, indicating if pairs of nodes are in the same branch, and we formulate decoding as a logic program. Each branch from the root can then be interpreted as a cluster containing all coreferent mentions of an entity and all quotes from that entity.

In addition, we designed an evaluation metric suitable for entity-level quotation attribution that takes into account informative speakers. Experimental results show mutual improvements in the coreference resolution and quotation attribution tasks.

Future work will include extensions to tackle indirect quotations, possibly exploring connections to semantic role labeling.

Acknowledgements

We thank all reviewers for their valuable comments, and Silvia Pareti and Tim O’Keefe for providing us the PARC dataset and answering several questions. This work was partially supported by the EU/FEDER programme, QREN/PORT Lisboa (Portugal), under the Intelligo project (contract 2012/24803) and by a FCT grant PTDC/EEI-SII/2312/2012.
References


David K Elson and Kathleen McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In AAAI.


