Model Architectures for Quotation Detection

Christian Scheible, Roman Klinger and **Sebastian Padó**
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
{scheibcn,klinger,pado}@ims.uni-stuttgart.de

Abstract

Quotation detection is the task of locating spans of quoted speech in text. The state of the art treats this problem as a sequence labeling task and employs linear-chain conditional random fields. We question the efficacy of this choice: The Markov assumption in the model prohibits it from making joint decisions about the begin, end, and internal context of a quotation. We perform an extensive analysis with two new model architectures. We find that (a), simple boundary classification combined with a greedy prediction strategy is competitive with the state of the art; (b), a semi-Markov model significantly outperforms all others, by relaxing the Markov assumption.

1 Introduction

Quotations are occurrences of reported speech, thought, and writing in text. They play an important role in computational linguistics and digital humanities, providing evidence for, e.g., speaker relationships (Elson et al., 2010), inter-speaker sentiment (Nalisnick and Baird, 2013) or politeness (Faruqui and Pado, 2012). Due to a lack of general-purpose automatic systems, such information is often obtained through manual annotation (e.g., Agarwal et al. (2012)), which is labor-intensive and costly. Thus, models for *automatic quotation detection* form a growing research area (e.g., Pouliquen et al. (2007); Pareti et al. (2013)).

Quotation detection looks deceptively simple, but is challenging, as the following example shows:

> [The pipeline], the company said, [would be built by a proposed joint venture ..., and Trunkline ... will "build and operate" the system ...].[1]

[1]Penn Attributions Relation Corpus (PARC), wsj_0260

Note that quotations can (i) be signalled by lexical cues (e.g., communication verbs) without quotation marks, (ii) contain misleading quotation marks; (iii) be discontinuous, and (iv) be arbitrarily long.

Early approaches to quotation detection use hand-crafted rules based on syntactic markers (Pouliquen et al., 2007; Krestel et al., 2008). While yielding high precision, they suffered from low recall. The state of the art (Pareti et al., 2013; Pareti, 2015) treats the task as a sequence classification problem and uses a linear-chain conditional random field (CRF). This approach works well for the prediction of the approximate location of quotations, but yields a lower performance detecting their exact span.

In this paper, we show that linear-chain sequence models are a sub-optimal choice for this task. The main reason is their *length*, as remarked above: Most sequence labeling tasks in NLP (such as most cases of named entity recognition) deal with spans of a few tokens. In contrast, the median quotation length on the Penn Attributions Relation Corpus (PARC, Pareti et al. (2013)) is 16 tokens and the longest span has over 100 tokens. As a result of the strong Markov assumptions that linear-chain CRFs make to ensure tractability, they cannot capture "global" properties of (almost all) quotations and are unable to make joint decisions about the begin point, end point, and content of quotations.

As our first main contribution in this paper, we propose two novel model architectures designed to investigate this claim. The first is *simpler* than the CRF. It uses token-level classifiers to predict quotation boundaries and combines the boundaries greedily to predict spans. The second model is *more expressive*. It is a semi-Markov sequence model which relaxes the Markov assumption, enabling it to consider global features of quotation spans. In our second main contribution, an analysis of the models' performances, we find that the sim-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1736–1745,
Berlin, Germany, August 7-12, 2016.

pler model is competitive with the state-of-the-art CRF. The semi-Markov model outperforms both of them significantly by 3 % F_1. This demonstrates that the relaxed Markov assumptions help improve performance. Our final contribution is to make implementations of all models publicly available.[2]

2 The Task: Quotation Detection

Problem Definition Following the terminology established by Pareti et al. (2013), we deal with the detection of *content spans*, the parts of the text that are being quoted. To locate such spans, it is helpful to first detect *cues* which often mark the beginning or end of a quotation. The following example shows an annotated sentence from the PARC corpus; each content span (CONT) is associated with exactly one cue span (CUE):

> Mr. Kaye [denies]$_{\text{CUE}}$ [the suit's charges]$_{\text{CONT}}$ and [says]$_{\text{CUE}}$ [his only mistake was taking on Sony in the marketplace]$_{\text{CONT}}$.[3]

Pareti et al. (2013) distinguish three types of quotations. *Direct* quotations are fully enclosed in quotation marks and are a verbatim reproduction of the original utterance. *Indirect* quotations paraphrase the original utterance and have no quotation marks. *Mixed* quotations contain both verbatim and paraphrase content and may thus contain quotation marks. Note that the type of a content span is assigned automatically based on its surface form using the definitions just given.

Quotation Detection as Sequence Modeling In this paper, we compare our new model architectures to the state-of-the-art approach by Pareti (2015), an extension of Pareti et al. (2013). Their system is a pipeline: Its first component is the *cue model*, a token-level k-NN classifier applied to the syntactic heads of all verb groups. After cues are detected, content spans are localized using the *content model*, a linear-chain conditional random field (CRF) which makes use of the location of cues in the document through features.

As their system is not publicly available, we reimplement it. Our cue classifier is an averaged perceptron (Collins, 2002) which we describe in more detail in the following section. It uses the features in Table 1.[4] Our content model is a CRF with BIOE labels. It uses all features from Table 1 plus features that build on the output of the cue classifier, shown in Table 2.

C1. Surface form, lemma, and PoS tag for all tokens within a window of ± 5.
C2. Bigrams of surface form, lemma, and PoS tag
C3. Shape of t_i
C4. Is any token in a window of ± 5 a named entity?
C5. Does a quotation mark open or close at t_i (determined by counting)? Is t_i within quotation marks?
C6. Is t_i in the list of reporting verbs, noun cue verbs, titles, WordNet persons or organizations, and its VerbNet class
C7. Do a sentence, paragraph, or the document begin or end at t_i, t_{i-1}, or t_{i+1}?
C8. Distance to sentence begin and end; sentence length
C9. Does the sentence contain t_i a pronoun/named entity/quotation mark?
C10. Does a syntactic constituent starts or ends at t_i?
C11. Level of t_i in the constituent tree
C12. Label and level of the highest constituent in the tree starting at t_i; label of t_i's the parent node
C13. Dependency relation with parent or any child of t_i (with and without parent surface form)
C14. Any conjunction of C5, C9, C10

Table 1: Cue detection features for a token t_i at position i, mostly derived from Pareti (2015)

S1. Is a direct or indirect dependency parent of t_i classified as a cue, in the cue list, or the phrase "according to"?
S2. Was any token in a window of ± 5 classified as a cue?
S3. Distance to the previous and next cue
S4. Does the sentence containing t_i have a cue?
S5. Conjunction of S4 and all features from C14

Table 2: Additional features for content span detection, mostly derived from Pareti (2015)

3 New Model Architectures

While Pareti (2015) apply sequence modeling for quotation detection, they do not provide an analysis what the model learns. In this paper, we follow the intuition that a linear-chain CRF mostly makes local decisions about spans, while ignoring their global structure, such as joint information about the context of the begin and end points. If this is true, then (a) a model might work as well as the CRF without learning from label sequences, and (b) a model which makes joint decisions with global information might improve over the CRF.

This motivates our two new model architectures for the task. We illustrate the way the different architectures make use of information in Figure 1. Our simpler model (GREEDY) makes strictly local classification decisions, completely ignoring

[2] http://www.ims.uni-stuttgart.de/data/qsample

[3] PARC, `wsj_2418`

[4] For replicability, we give more detailed definitions of the features in the supplementary notes.

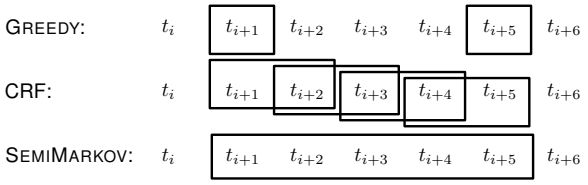

Figure 1: Information usage by model architecture. Frames indicate joint decisions on token labels.

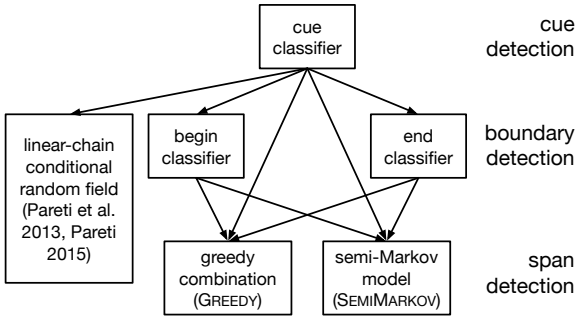

Figure 2: Information flow in all three models

those around it. The CRF is able to coordinate decisions within a window, which is propagated through Viterbi decoding. The more powerful model (SEMIMARKOV) takes the full span into account and makes a joint decision about the begin and end points.

Our intuition about the shortcomings of the CRF is based on an empirical analysis. However, to simplify the presentation, we postpone the presentation of this analysis to Section 6 where we can discuss and compare the results of all three models.

3.1 Model Decomposition and Formalization

We first introduce a common formalization for our model descriptions. Our problem of interest is *content span detection*, the task of predicting a set $\mathcal{S}$ of content spans (t_b, t_e) delimited by their begin and end tokens. The CRF solves this task by classifying tokens as begin/end/inside/outside tokens and thus solves a proxy problem. The problem is difficult because corresponding begin and end points need to be matched up over long distances, a challenge for probabilistic finite state automata such as CRFs.

In our model, *cue detection*, the task of detecting cue tokens t_c (cf. Section 2), remains the first step. However, we then decompose the content span problem solved by the CRF by introducing the intermediary task of *boundary detection*. As illustrated in Figure 2, this means identifying the sets of all *begin* and *end* tokens, t_b and t_e, ignoring their interdependencies. We then recombine these

Algorithm 1 GREEDY content span algorithm

Input: List of documents D; feature functions $\boldsymbol{f}_x$ for cue, begin, and end ($x \in c, b, e$); distance parameter $d_{\max}$; length parameter $\ell_{\max}$
Output: Content span labeling $\mathcal{S}$

$\boldsymbol{\theta}_c, \boldsymbol{\theta}_b, \boldsymbol{\theta}_e \leftarrow$ TRAINCLASSIFIERS$(D, \boldsymbol{f}_c, \boldsymbol{f}_b, \boldsymbol{f}_e)$
for d in D **do**
  $\mathcal{S} \leftarrow \varnothing$
  for token t in d **do**
    if $\boldsymbol{\theta}_c \cdot \boldsymbol{f}_c(t) > 0$ **then**
      $t_b \leftarrow$ next token right of t where $\boldsymbol{\theta}_b \cdot \boldsymbol{f}_b(t) > 0$ ▷ next begin
      $t_e \leftarrow$ next token right of t_b where $\boldsymbol{\theta}_e \cdot \boldsymbol{f}_e(t) > 0$ ▷ next end
      if $|t_b - t_c| \leq d_{\max}$ **and** $|t_e - t_b| \leq \ell_{\max}$ **and** OVERLAPPING$(t_b, t_e) = \varnothing$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(t_b, t_e)\}$ ▷ add span

predictions with two different strategies, as detailed in Section 3.2 and Section 3.3. This decomposition has two advantages: (a), we expect that boundary detection is easier than content span detection, as we remove the combinatorial complexity of matching begin and end tokens; (b), begin, end, and cue detection are now three identical classification tasks that can be solved by the same machinery.

We model each of the three tasks (cue/begin/end detection) with a linear classifier of the form

$$\text{score}_x(t) = \boldsymbol{\theta}_x \cdot \boldsymbol{f}_x(t) \quad (1)$$

for a token t, a class $x \in \{c, b, e\}$ (for *cue*, *begin*, and *end*), a feature extraction function $\boldsymbol{f}_x(t)$, and a weight vector $\boldsymbol{\theta}_x$. We re-use the feature templates from Section 2 to remain comparable to the CRF.

We estimate all parameters $\boldsymbol{\theta}_x$ with the perceptron algorithm, and use parameter averaging (Collins, 2002). Since class imbalances, which occur in the boundary detection tasks, can have strong effects (Barandela et al., 2003), we train the perceptron with uneven margins (Li et al., 2002). This variant introduces two learning margins: τ_{-1} for the negative class and τ_{+1} for the positive class. Increasing τ_{+1} at a constant τ_{-1} increases recall (as failure to predict this class is punished more), potentially at the loss of precision, and vice versa.

3.2 Greedy Span Detection

Our first new model, GREEDY (Figure 2, bottom center), builds on the assumption that the modeling of sequence properties in a linear-chain CRF is weak enough that sequence learning can be replaced by a greedy procedure. Algorithm 1 shows how we generate a span labeling based on the output of the boundary classifiers. Starting at each cue,

we add all spans within a given distance $d_{\max}$ from the cue whose length is below a given maximum $\ell_{\max}$. If the candidate span is OVERLAPPING with any existing spans, we discard it. Analogously, we search for spans to the left of the cue. The algorithm is motivated by the structure of attribution relations: each content span has one associated cue.

3.3 Semi-Markov Span Detection

Our second model extends the CRF into a semi-Markov architecture which is able to handle *global features* of quotation span candidates (SEMIMARKOV, Figure 2 bottom right). Following previous work (Sarawagi and Cohen, 2004), we relax the Markov assumption inside spans. This allows for extracting arbitrary features on each span, such as conjunctions of features on the begin and end tokens or occurrence counts within the span.

Unfortunately, the more powerful model architecture comes at the cost of a more difficult prediction problem. Sarawagi and Cohen (2004) propose a variant of the Viterbi algorithm. This however does not scale to our application, since the maximum length of a span factors into the prediction runtime, and quotations can be arbitrarily long. As an alternative, we propose a sampling-based approach: we draw candidate spans (*proposals*) from an informed, non-uniform distribution of spans. We score these spans to decide whether they should be added to the document (*accepted*) or not (*rejected*). This way, we efficiently traverse the space of potential span assignments while still being able to make informed decisions (cf. Wick et al. (2011)).

To obtain a distribution over spans, we adapt the approach by Zhang et al. (2015). We introduce two independent probability distributions: P_b is the distribution of probabilities of a token being a *begin token*; P_e is the distribution of probabilities of a token being an *end token*. We sample a single content span proposal (DRAWPROPOSAL) by first sampling the *order* in which the boundaries are to be determined (begin token or end token first) by sampling a binary variable $d \sim \text{Bernoulli}(0.5)$. If the begin token is to be sampled first, we continue by drawing a begin token $t_b \sim P_b$ and finally draw an end token $t_e \sim P_e$ within a window of up to $\ell_{\max}$ tokens to the right of t_b. If the end token is to be sampled first, we proceed conversely. We also propose empty spans, i.e., the removal of existing spans without an replacement.

For the distributions P_b and P_e, we reuse our boundary detection models from Section 3.1. For each class $x \in \{b, e\}$ we form a distribution

$$P_x(t) \propto \exp(\text{score}_x(t)/T_x) \quad (2)$$

over the tokens t of a document using the scores from Equation 1. T_x is a temperature hyperparameter. Temperature controls the pronouncedness of peaks in the distribution. Higher temperature flattens the distribution and encourages the selection of tokens with lower scores. This is useful when exploration of the sample space is desired.

Algorithm 2 SEMIMARKOV inference algorithm

Input: Document d; probability distributions for begin and end (P_b, P_e); feature function for spans $\boldsymbol{g}$; maximum span length $\ell_{\max}$; number of proposals N
Output: Set of content spans $\mathcal{S}$

$\mathcal{S} \leftarrow \varnothing$
$\boldsymbol{\theta} \leftarrow \mathbf{0}$
for $n = 1$ to N **do**
  $(t_b, t_e) \leftarrow$ DRAWPROPOSAL(P_b, P_e)
  score $\leftarrow \boldsymbol{\theta} \cdot \boldsymbol{g}(t_b, t_e)$
  $\mathcal{O} \leftarrow$ OVERLAPPING(t_b, t_e)
  score$_{\mathcal{O}} \leftarrow \sum_{(t'_b, t'_e) \in \mathcal{O}} \boldsymbol{\theta} \cdot \boldsymbol{g}(t'_b, t'_e)$
  if score $>$ score$_{\mathcal{O}}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{O}$ ▷ remove overlapping
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(t_b, t_e)\}$ ▷ accept proposal
    if ISTRAINING **and** $\neg$CORRECT(t_b, t_e) **then**
      PERCEPTRONUPDATE ▷ wrongly accepted
  else
    REJECT(t_b, t_e)
    if ISTRAINING **and** CORRECT(t_b, t_e) **then**
      PERCEPTRONUPDATE ▷ wrongly rejected

The proposed candidates enter into the decision algorithm shown in Algorithm 2. As shown, the candidates are scored using a linear model (again as defined in Equation 1). We use the features of the previous models (Table 1 and 2) on the begin and end tokens. As we now judge complete span assignments rather than local label assignments to tokens, we can add a new span-global feature function $\boldsymbol{g}(t_b, t_e)$. We introduce the features shown in Table 3. If the candidate's score is higher than the sum of scores of all spans overlapping with it, we accept it and remove all overlapping ones.

This model architecture can be seen as a modification of the pipeline of the GREEDY model (cf. Figure 2). We again detect cues and boundaries, but then make an informed decision for combining begin and end candidates. In addition, the sampler makes "soft" selections of begin and end tokens based on the model scores rather than simply accepting the classifier decisions.

For training, we again use perceptron updates (cf. Section 3.2). If the model accepts a wrong

	Setting	Direct			Indirect			Mixed			Overall		
		P	R	F	P	R	F	P	R	F	P	R	F
strict	Pareti (2015) as reported therein	94	88	91	78	56	65	67	60	63	80	63	71
	CRF (own re-implementation)	94	93	94	73	58	64	81	68	74	79_{g}	67	72
	GREEDY	92	91	91	69	59	64	72	64	68	75	67	71
	SEMIMARKOV	93	94	94	73	65	69	81	66	73	$\mathbf{79}_{g}$	$\mathbf{71}^{c}_{g}$	$\mathbf{75}^{c}_{g}$
	Combination: CRF+SEMIMARKOV	94	93	94	73	64	69	81	68	74	$\mathbf{79}_{g}$	$\mathbf{71}^{c}_{g}$	$\mathbf{75}^{c}_{g}$
partial	Pareti (2015) as reported therein	99	93	96	91	66	77	91	81	86	93	73	82
	CRF (own re-implementation)	98	96	97	87	70	77	94	83	88	90_{g}	77	83
	GREEDY	97	95	96	83	76	79	93	85	89	88	81^{c}	84
	SEMIMARKOV	97	95	96	83	75	79	92	81	86	88	80	84
	Combination: CRF+SEMIMARKOV	98	96	97	83	75	79	94	83	88	88	81^{c}	84^{c}

Table 4: Results on the test set of PARC3. Best overall strict results in bold. Models as in Figure 2. g: significantly better than GREEDY; c: significantly better than CRF (both with $\alpha = 0.05$).

G1. Numbers of named entities, lowercased tokens, commas, and pronouns inside the span
G2. Binned percentage of tokens that depend on a cue
G3. Location of the closest cue (left/right?), percentage of dependents on that cue
G4. Number of cues overlapped by the span
G5. Is there a cue before the first token and/or after the last token of the span (within the same sentence)? first or after the last token of the span?, and their conjunction
G6. Do both the first and the last token depend on a cue?
G7. Binned length of the span
G8. Does the span match a sentence exactly/off by one token?
G9. Number of sentences covered by the span
G10. Does the span match one or more constituents exactly?
G11. Is the span direct, indirect, or mixed?
G12. Is the # of quotation marks in the span odd or even?
G13. Is the span is direct and does it contain more than two quotation marks?

Table 3: Global features for content span detection

span, we perform a negative update (Line 12 in Algorithm 2). If a correct span is rejected, we make a positive update (Line 16). We iterate over the documents in random order for a fixed number E of epochs. As the sampling procedure takes long to fully label documents, we employ GREEDY to make initial assignments. This does not constitute additional supervision, as the sampler can remove any initial span and thus refute the initialization. This reduces runtime without affecting the result in practice.

4 Experimental Setup

Data We use the Penn Attribution Relations Corpus, version 3 (henceforth *PARC3*), by Pareti (2015).[5] It contains AR annotations on the Wall Street Journal part of the Penn Treebank (2,294 news documents). As in related work, we use sections 1–22 as training set, section 23 as test set, and section 24 as development set. We perform the same preprocessing as Pareti: We use gold tokenization, lemmatization, part-of-speech tags, constituency parses, gold named entity annotations (Weischedel and Brunstein, 2005), and Stanford parser dependency analyses (Manning et al., 2014).

Evaluation We report precision, recall, and micro-averaged F_1, adopting the two metrics introduced by Pareti et al. (2013): *Strict match* considers cases as correct where the boundaries of the spans match exactly. *Partial match* measures correctness as the ratio of overlap of the predicted and true spans. In both cases, we report numbers for each of the three quotation types (*direct*, *indirect*, *mixed*) and their micro averages. Like Pareti (2015), we exclude single-token content spans from the evaluation. To test for statistical significance of differences, we use the approximate randomization test (Noreen, 1989) at a significance level of $\alpha = 0.05$.

Implementation and Hyperparameters We use the CRF implementation in MALLET (McCallum, 2002). We optimize all hyperparameters of the models on the development set. Our best models use positive margins of $\tau_+ = 25$ for the boundary and $\tau_+ = 15$ for the span models, favoring recall. The SEMIMARKOV sampler uses a temperature of $T_x = 10$ for all classes. We perform 15 epochs of training after which the models have converged, and draw 1,000 samples for each document. For the GREEDY model, we obtain the best results with $d_{\max} = 30$ and $\ell_{\max} = 55$. For the SEMIMARKOV sampler, $\ell_{\max} = 75$ is optimal.

[5] Note that the data and thus the results differ from those previously published in (Pareti et al., 2013).

The high values mirror the presence of very long spans in the data.

5 Results

Cue We first evaluate the cue classifier. We obtain an F_1 of 86 %, with both precision and recall at 86 %, which is very close to the 85 % F_1 of Pareti.

CRF Table 4 summarizes the content span results. First, we compare Pareti's results to our reimplementation (the rows denoted with *Pareti (2015)* and *CRF*). There are some differences in how well the model performs on certain types of spans: while our precision is lower for indirect spans, it is higher on mixed spans. Additionally, our implementation generally has higher recall than Pareti's. Her system includes several features using proprietary lists (such as a manually curated list of titles) we were unable to obtain, and complex feature templates that we may interpret differently. We suspect that these differences are due to the typical replication problems in NLP (cf. Fokkens et al. (2013)). Overall, however, our model performs quite similarly to Pareti's, with our model scoring an overall F_1 of 72 % (vs. Pareti's 71 %) and a partial F_1 of 83 % (vs. 82 %).

GREEDY Next, we compare the GREEDY model to the CRF. We find its overall performance to be comparable to the CRF, confirming our expectations. While strict precision is statistically significantly lower for GREEDY (75 % vs. 79 %), strict recall is not significantly different (bot at 67 %). Considering partial matches, GREEDY has significantly higher recall (81 % vs. 77 %) but significantly lower precision (88 % vs. 90 %) than the CRF, with an overall comparable F_1. This result bolsters our hypothesis that the CRF learn only a small amount of useful sequence information. Although GREEDY ignores label sequences in training completely, it is able to compete with the CRF. Furthermore, the partial match result that GREEDY is a particularly good choice if the main interest is the approximate location of content spans in a document: The simpler model architecture makes it easier and more efficient to train and apply. The caveat is that GREEDY is particularly bad at locating mixed spans (as indicated by a precision of only 72 %): Quotation marks are generally good indicators for span boundaries and are often returned as false positives by the boundary detection models, so GREEDY tends to incorrectly pick them.

SEMIMARKOV Overall, the SEMIMARKOV model outperforms the CRF significantly in terms of strict recall (71 % vs. 67 %) and F_1 (75 % vs. 72 %), while precision remains unaffected (at 79 %). The model performs particularly well on indirect quotations (increasing F_1 by 5 points to 69 %), the most difficult category, where local context is insufficient. Meanwhile, on partial match, the SEMIMARKOV model has a comparable recall (80 vs. 77 %), but significantly lower precision (88 % vs. 90 %). The overall partial F_1 results are not significantly different. The improvement on the strict measures supports our intuition that better features help in particular in identifying the *exact* boundaries of quotations, a task that evidently profits from global information.

Model Combination The complementary strengths of the CRF and SEMIMARKOV (CRF detects direct quotations well, SEMIMARKOV indirect quotations) suggest a simple model combination algorithm based on the surface form of the spans: First take all *direct* and *mixed* spans predicted by the CRF; then add all *indirect* spans from the SEMIMARKOV model (except for those which would overlap). This result is our overall best model under strict evaluation, although it is not significantly better than the SEMIMARKOV model. Considering partial match, its results are essentially identical to the SEMIMARKOV model.

6 Analysis

We now proceed to a more detailed analysis of the performance of the three models (CRF, GREEDY, and SEMIMARKOV) and their differences in order to gain insights into the nature of the quotation detection task. In the interest of readability, we organize this section by major findings instead of the actual analyses that we have performed, and adduce for each finding all relevant analysis results.

Finding 1: Variation in length does not explain the differences in model performance. A possible intuition about our models it that the improvement of SEMIMARKOV over CRF is due to a better handling of longer quotations. However, this is not the case. Figure 3 shows the recall of the three models for quotations binned by lengths. The main patterns hold across all three models: Medium-length spans are the easiest to detect. Short spans are difficult to detect as they are often part of discontinuous content spans. Long spans are also

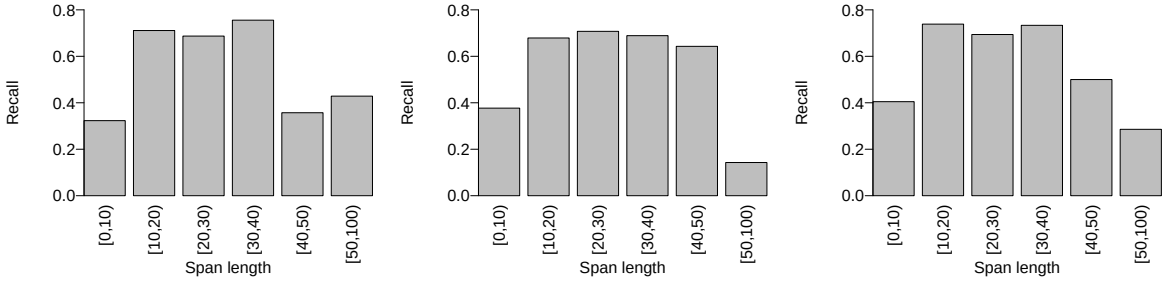

Figure 3: Strict recall by span length for CRF (left), GREEDY (center), and SEMIMARKOV model (right)

Category	Count		
	B	I	E
looking left	27	14	7
looking right	1	13	30
cue	11	10	7
other lexical	31	21	22
structural/syntactic	27	44	35
punctuation	31	25	36

Table 5: Categories of top positive and negative CRF features for begin (B), inside (I), and end (E)

difficult since any wrong intermediary decision can falsify the prediction. In fact, the CRF model is even the best model among the three for very long spans (which are rare). Those spans exceed the 55 and 75 token limits $\ell_{\max}$ of the GREEDY and SEMIMARKOV models. Intuitively, for the CRF, most spans are long: even spans which are short in comparison to other quotations are longer than the window within which the CRF operates. This is why span length does not have an influence.

Finding 2: Quotations are mostly defined by their immediate external context. A feature analysis of the CRF model reveals that many important features refer to material *outside* the quotation itself. For each label (B, I, E), we collect the 50 features with the highest positive and negative values, respectively. We first identify the subset of those features that looks look left or right. As the upper part of Table 5 shows, a substantial number of B (begin) features look to the left, and a number of E (end) features look to the right. Thus, these features do not look at the quotation itself, but at its immediate external context.

We next divide the features into four broad categories (cues, other lexical information, structural and syntactic features, and punctuation including quotation marks). The results in the lower part of Table 5 show that the begin and end classes rely on a range of categories, including lexical, cue and punctuation *outside* the quotation. The situation is different for inside tokens (I), where most features express structural and syntactic properties of the quotation such as the length of a sentence and its syntactic relation to a cue. Together, these observations suggest that one crucial piece of information about quotations is their lexical and orthographic context: the factors that mark a quotation as a quotation. Another crucial piece are internal structural properties of the quotation, while lexical properties of the quotation are not very important: which makes sense, since almost anything can be quoted.

The feature analysis is bolstered by an error analysis of the false negatives in the high-precision low-recall CRF. The first reason for false negatives is indeed the occurrence of infrequent cues which the cue model fails to identify (e.g., *read* or *acknowledge*). The second one is that the model does attempt to learn syntactic features, but that the structural features that can be learned by the CRF (such as C7, C10 or S4) can model only local windows of the quality of the quotation, but not its global quality. This leads us to our third finding.

Finding 3: Simple models cannot capture dependencies between begin and end boundaries well. Given the importance of cues, as evidenced by our Finding 2, we can ask whether the boundary of the quotation that is adjacent to its associated cue ("cue-near") is easier to identify than the other boundary ("cue-far") whose context is less informative. To assess this question, we evaluate the recall of individual boundary detection at the token level. For the CRF, "cue-far" boundaries of spans indeed tend to be more difficult to detect than "cue-near" ones. The results in Table 6 show that both the GREEDY and the CRF model show a marked asym-

	GREEDY	CRF	SEMIMARKOV
cue-near	76	74	76
cue-far	72	71	75

Table 6: Recall on boundaries by cue position

metry and perform considerably worse (3 % and 4 %, respectively) on the cue-far boundary. This asymmetry is considerably weaker for the SEMIMARKOV model, where both boundary types are recognized almost on par. The reason behind this finding is that neither the GREEDY model nor the CRF can condition the choice of the cue-far boundary on the cue-near boundary or on global properties of the quotation – the GREEDY model, because its choices are completely independent, and the CRF model, because its choices are largely independent due to the Markov assumption.

Finding 4: The SEMIMARKOV model benefits the most from its ability to handle global features about content spans. This leads us to our final finding about why the SEMIMARKOV model outperforms the CRF – whether it is the model architecture itself, or the new global features that it allows us to formulate. We perform an ablation study whose results are shown in Figure 4. We begin with only the token-level features on the begin, end, and interior tokens of the span, as introduced in Section 2, i.e., the features that the CRF has at its disposal. We find that this model performs on par with the CRF, thus the model architecture on its own does not help. We then incrementally add the feature templates containing count statistics of the internal tokens (Template G1 in Table 3) and advanced cue information (G2–G6). Both give the model incremental boosts. Adding syntactic coherence features (G7–G13) completes our full feature set and yields the best results.

Thus, the difference comes from features that describe global properties of the quotation. One of the most informative (negative) features is the conjunction from G6. It enforces the constraint that each content span is associated with a single cue. As in the CRF, the actual content of a content span does not play a large role. The only semantic features the model considers concern the presence of named entities within the span.

These observations are completed by analysis of the quotation spans that were correctly detected by the SEMIMARKOV model, but not the CRF (in terms of strict recall). We find a large amount of spans with highly ambiguous cue-near tokens such as *to* (10 % of the cases) *that* (16 %). We find that often the errors are also related to the frequency or location of cues. As an example, in the sentence

> [...] he has said [that when he was on the winning side in the 1960s, he knew that the tables might turn in the future]$_{\text{CONT}}$.[6]

the CRF model incorrectly splits the content span at the second cue candidate *knew*. This is, however, an embedded quotation that the model should ignore. In contrast, the SEMIMARKOV model makes use of the fact the tokens of the span depend on the same cue, and predicts the span correctly. For these tokens, the distinction between reported speech and factual descriptions is difficult. Arguably, it is the global features that help the model make its call.

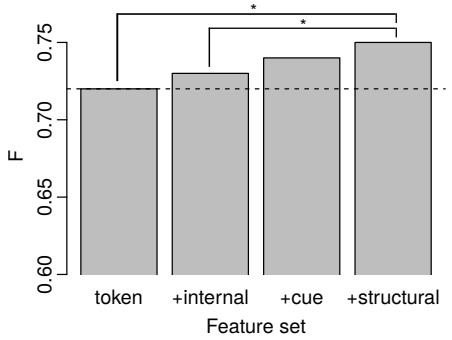

Figure 4: Strict F_1 for different feature sets in the SEMIMARKOV model. *: Difference statistically significant. Dashed line: CRF result.

7 Related Work

Quotation detection has been tackled with a number of different strategies. Pouliquen et al. (2007) use a small set of rules which has high precision but low recall on multilingual text. Krestel et al. (2008) also pursue a rule-based approach, focusing on the roles of cue verbs and syntactic markers. They evaluate on a small set of annotated WSJ documents and again report high precision but low recall. Pareti et al. (2013) develop the state-of-the-art sequence labeling approach discussed in this paper.

Our sampling approach builds on that of Zhang et al. (2015), who pursue a similar strategy for parsing, PoS tagging, and sentence segmentation. Similar semi-Markov model approaches have been used for other applications, e.g. by Yang and Cardie

[6] PARC, `wsj_2347`

(2012) and Klinger and Cimiano (2013) for sentiment analysis. They also predict spans by sampling, but they draw proposals based on the token or syntactic level. This is not suitable for quotation detection as we deal with much longer spans.

8 Conclusion

We have considered the task of quotation detection, starting from the hypothesis that linear-chain CRFs cannot take advantage of all available sequence information due to its Markov assumptions. Indeed, our analyses find that the features most important to recognize a quotation consider its direct context of orthographic evidence (such as quotation marks) and lexical evidence (such as cue words). A simple, greedy algorithm using non-sequential models of quotation boundaries rivals the CRF's performance. For further improvements, we introduce a semi-Markov model capable of taking into account *global* information about the complete span not available to a linear-chain CRF, such as the presence of cues on both sides of the quotation candidate. This leads to a significant improvement of 3 points F_1 over the state of the art.

On a more general level, we believe that quotation detection is interesting as a representative of tasks involving long sequences, where Markov assumptions become inappropriate. Other examples of such tasks include the identification of chemical compound names (Krallinger et al., 2015) and the detection of annotator rationales (Zaidan and Eisner, 2008). We have shown that a more expressive semi-Markov model which avoids these assumptions can improve performance. More expressive models however come with harder inference problems which are compounded when applied to long-sequence tasks. The informed sampling algorithm we have described performs such efficient inference for our semi-Markov quotation detection model.

Acknowledgments

This work was funded in part by the DFG through the Sonderforschungsbereich 732. We thank Silvia Pareti for kindly providing the PARC dataset as well as for much information helpful for replicating her results. Further thanks go to Anders Björkelund and Kyle Richardson for discussion and comments.

References

Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. 2012. Social network analysis of alice in wonderland. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 88–96, Montréal, Canada, June. Association for Computational Linguistics.

Ricardo Barandela, José Salvador Sánchez, Vicente García, and Edgar Rangel. 2003. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Philadelphia, PA.

David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden.

Manaal Faruqui and Sebastian Pado. 2012. Towards a model of formal and informal address in English. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 623–633, Avignon, France.

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1691–1701, Sofia, Bulgaria.

Roman Klinger and Philipp Cimiano. 2013. Bidirectional inter-dependencies of subjective expressions and targets and their value for a joint model. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 848–854, Sofia, Bulgaria.

Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. 2015. CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, 7(Suppl 1):S1.

Ralf Krestel, Sabine Bergler, and René Witte. 2008. Minding the source: Automatic tagging of reported speech in newspaper articles. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 2823–2828, Marrakech, Morocco.

Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz S. Kandola. 2002. The perceptron algorithm with uneven margins. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 379–386, Sydney, Australia.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL System Demonstrations*, pages 55–60, Baltimore, MD.

Andrew K. McCallum, 2002. *MALLET: A Machine Learning for Language Toolkit*. User's manual.

Eric T. Nalisnick and Henry S. Baird. 2013. Character-to-character sentiment analysis in shakespeare's plays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 479–483, Sofia, Bulgaria, August. Association for Computational Linguistics.

Eric W. Noreen. 1989. *Computer intensive methods for hypothesis testing: An introduction*. Wiley, New York.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 989–999, Seattle, WA.

Silvia Pareti. 2015. *Attribution: A Computational Approach*. Ph.D. thesis, University of Edinburgh.

Bruno Pouliquen, Ralf Steinberger, and Clive Best. 2007. Automatic detection of quotations in multilingual news. In *Proceedings of Recent Advances in Natural Language Processing*, pages 487–492, Borovets, Bulgaria.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1185–1192, Vancouver, BC.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. Linguistic Data Consortium, Philadelphia.

Michael Wick, Khashayar Rohanimanesh, Kedar Bellare, Aron Culotta, and Andrew McCallum. 2011. Samplerank: Training factor graphs with atomic gradients. In *Proceedings of the 28th International Conference on Machine Learning*, pages 777–784, Bellevue, WA.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-Markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1335–1345, Jeju Island, South Korea.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40, Honolulu, HI.

Yuan Zhang, Chengtao Li, Regina Barzilay, and Kareem Darwish. 2015. Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 42–52, Denver, CO.