Direct Speech Quote Attribution for Dutch Literature

Andreas van Cranenburgh
Center for Language and Cognition
University of Groningen
`a.w.van.cranenburgh@rug.nl`

Frank van den Berg
Klippa
Groningen, The Netherlands
`frankvandenberg@klippa.com`

Abstract

We present a dataset and system for quote attribution in Dutch literature. The system is implemented as a neural module in an existing NLP pipeline for Dutch literature (dutchcoref; van Cranenburgh, 2019). Our contributions are as follows. First, we provide guidelines for Dutch quote attribution and annotate 3,056 quotes in fragments of 42 Dutch literary novels, both contemporary and classic. Second, we present three neural quote attribution classifiers, optimizing for precision, recall, and F1. Third, we perform an evaluation and analysis of quote attribution performance, showing that in particular, quotes with an implicit speaker are challenging, and that such quotes are prevalent in contemporary fiction (57%, compared to 32% for classic novels). On the task of quote attribution, we achieve an improvement over the rule-based baseline of 8.0% F1 points on contemporary fiction and 1.9% F1 points on classic novels. Code, models, and annotations for the public domain novels are available under an open license at `https://github.com/frenkvdberg/dutchqa`.

1 Introduction

Quote attribution is the task of identifying the speaker of each quotation span in a given text. When applied to dialogue in literature, this enables us to study relations and interactions between characters, for example by extracting social networks (Elson et al., 2010; Labatut and Bost, 2019). Other applications to literature include examining gender differences (Underwood et al., 2018; Kraicer and Piper, 2019) or measuring information propagation (Sims and Bamman, 2020). Whereas the aforementioned studies all focus on English-language fiction, in this paper we focus on direct speech attribution in Dutch literature. An example can be seen in the following sentence:

(1) *“[Ik]$_1$ denk dat [je]$_2$ met [haar]$_3$ moet praten,” zei [Tom]$_1$.* Speaker: [Tom]$_1$
“[I]$_1$ think [you]$_2$ should talk to [her]$_3$”, [Tom]$_1$ said.

Example (1) has an **explicit** speaker mention (*Tom*). However, identifying the speaker is not always as easy (examples from Tolstoy’s Anna Karenina):

(2) *“Maar, wat nu te doen?” vroeg [hij]$_1$ wanhopig.* Speaker: [Stepan]$_1$
“Well, what now?” [he]$_1$ asked disconsolately.

(3) *“[Mama]$_1$?, [ze]$_1$ is opgestaan,” antwoordde [het meisje]$_2$.* Speaker: [Tanya]$_2$
“[Mamma]$_1$? [She]$_1$ is up,” answered [the girl]$_2$.

(4) *“Het komt goed, [meneer]$_1$; [ze]$_2$ draait wel bij,” zei [Matvey]$_3$.* Speaker: [Matvey]$_3$
“Bijdraaien?” Speaker: [Stepan]$_1$
“Ja, [meneer]$_1$.” Speaker: [Matvey]$_3$
It’s all right, [sir]$_1$; [she]$_2$ will come round,” said [Matvey]$_3$.
“Come round?”
“Yes, [sir]$_1$.”

The speakers of (2) and (3) are mentioned **by anaphor**. Sentences 2–3 of (4) are even more challenging, since they have an **implicit** speaker. Note that all of the above examples are direct speech, in which the exact words spoken by a person are reported. Although there exist systems for detecting and attributing indirect speech (Pareti et al., 2013; Salway et al., 2017) and free indirect discourse (Brooke et al., 2017), our focus in this work is strictly on direct speech.

For the task of Dutch quote attribution, van Cranenburgh (2019) presents a rule-based approach as part of the *dutchcoref* coreference resolution system. Quote attribution is relevant to coreference resolution, as the speaker and addressee of dialogue turns must be known to resolve first and second person pronouns in quoted speech correctly. Furthermore, after extending the dutchcoref system with three neural classifiers for the subtasks mention de-

Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL2023), pages 45–62
May 5, 2023 ©2023 Association for Computational Linguistics

tection, mention attributes and pronoun resolution, van Cranenburgh et al. (2021) notes that in literature dialogue is particularly important; annotating and predicting speakers of direct speech was proposed as one of the directions for future work. Therefore, we implement a neural classifier for quote attribution, which we expect to outperform dutchcoref's rule-based approach.

Additionally, we perform an error analysis, where we look at whether certain speaker types are harder to classify: **explicit** (said Tom) vs. **anaphoric pronoun** (said he) vs. **anaphoric other** (said his friend) vs. **implicit**. Moreover, we will analyze whether the corpus of books that we use (RiddleCoref vs. OpenBoek) has an influence on the performance of the classifier.

2 Background

2.1 Quote attribution in the literary domain

Semino and Short (2004) present a taxonomy of speech and thought representation, and a corpus annotated with this taxonomy that includes fiction. Later work attempts to automate quote attribution. Glass and Bangay (2007) approach the task of quote attribution in the literary domain by combining a scoring technique and hand-coded rules to identify the speaker of quoted speech in fiction books. Their approach consists of three steps: identifying the speech verb for a quote, finding the actor for this speech verb and then selecting the correct speaker from a character list. While performing well, their system is limited to explicitly cued speakers and not able to identify implicit speakers. Elson and McKeown (2010) aim to automatically identify both quotes and the mentions of the speakers in a self compiled corpus of classic literature. However, their predictions rely on gold-label information at test time, which is not available in practice. O'Keefe et al. (2012) uses a sequence labeling approach, which proves successful for the news domain, but does not manage to beat their baseline accuracy on the literary domain. Subsequently, O'Keefe et al. (2013) reported on the impact of coreference resolution on the task of quote attribution, with Almeida et al. (2014) presenting a joint model of coreference resolution and quote attribution. Around the same time, the best system for literary quote attribution was the system by He et al. (2013), presenting a supervised machine learning approach. Instead of seeing the task as quote-mention labeling, they reformulated it to quote-speaker labeling. Their system was eventually outperformed by Muzny et al. (2017), who present a rule-based and statistical quote attribution system. Adding a supervised classifier to their deterministic sieve-based system proved successful on English literature, achieving an average F1-score of 87.5% across three novels. Yeung and Lee (2017) present a machine learning system that identifies not just the speaker of dialogue in literature, but also the addressee. Sims and Bamman (2020) reimplements the deterministic approach of Muzny et al. (2017), while also using coreference information and choosing to assign unattributed quotes to the majority speaker. Instead of evaluating system performance using accuracy and precision/recall, they measure the cluster overlap. Their system achieves an average F1-score of 71.3% across three different cluster metrics, when evaluated on their new dataset containing 1,765 quotes across 100 different literary texts. Byszuk et al. (2020) present an evaluation of direct speech attribution for 19th-century fiction in 9 languages by fine-tuning transformer-based sequence labeling models, which appear to be more robust to varying typographical conventions compared to rule-based approaches. Papay and Padó (2020) present a corpus of 19th century literature with rich dialogue annotations: direct and indirect speech are included, and not only speakers but also addressees and cue words are annotated. Yoder et al. (2021) present a neural pipeline tailored to English fan fiction, including character identification, coreference, and quote attribution. Most recently, Vishnubhotla et al. (2022) presented a dataset of all quotations in 22 English novels, with annotations for speaker, addressee, and other attributes. They also evaluate systems based on Muzny et al. (2017) and BookNLP,[1] and report lower scores (overall accuracy up to 63%) than with previous datasets, suggesting the task is more challenging than previously thought.

2.2 Quote attribution within dutchcoref

The dutchcoref system (van Cranenburgh, 2019; van Cranenburgh et al., 2021) performs quote attribution as part of its rule-based coreference resolution system, which follows the deterministic multi-sieve architecture of Lee et al. (2013). The system starts by identifying mention spans and attributes (animacy, gender, number). This is fol-

[1] BookNLP is a neural pipeline optimized for literature, cf. `https://github.com/booknlp/booknlp`

lowed by quote attribution and a sequence of rule-based sieves that make coreference decisions, ordered from most to least precise.

In the quote attribution component, direct speech is identified using punctuation: single and double quotation marks, and paragraphs that start with a dash. While this heuristic works for the majority of cases, there are rare cases where quotation marks are used for other things than direct speech. If no marker is found to indicate the end of direct speech, the system assumes the end of the paragraph is also the end of the quote. Furthermore, the system does not extract quotes within other quotes.

Speakers of direct speech are attributed where they are explicitly mentioned, such as when the subject of a reported speech verb is located next to a quote. Addressees are identified as well, as the addressee is set to the speaker of the previous or following quote. The system uses paragraph breaks in order to decide whether a speaker continues speaking or another participant takes the next turn. Even in a longer chain of implicit quotes, the system can still attribute the speakers and addressees, assuming that the same speaker pair keeps taking turns. Other heuristic rules for identifying speakers and addressees include recognizing certain vocative patterns and checking whether there is only a single human mention in the paragraph. These heuristic rules are similar to those reported in the paper by Muzny et al. (2017), although they do not discuss the identification of addressees.

The performance of dutchcoref's quote attribution component was reported using the first 1,000 quotes from the novel *De Buurman* by J.J. Voskuil. A low recall score of 43.3% was obtained, as almost half of the quotes were not assigned a speaker. However, the obtained precision score was high, scoring 81.7%. The low recall score can be explained by the decision to not assign unattributed quotes to the majority speaker, since the system was designed to favor precision. The error analysis revealed that most errors occurred where the speaker was implicit, with the quote attribution rules working well when speakers were mentioned explicitly.

As Muzny et al. (2017) obtained better results when combining the heuristic rules with a lightweight supervised classifier, a similar experiment was also tried for dutchcoref. van Cranenburgh (2019) trained a fastText classifier (Joulin et al., 2017) to classify the unattributed quotes, but the results were not encouraging, as there was not enough annotated data.

Seeing how a lack of training data caused the performance to be poor, we are curious how well a classifier can perform when we supply a sufficient amount of training data. Therefore, we annotated quotes appearing in fragments of 42 different Dutch-language novels and train our own classifier, which can then be implemented into the dutchcoref system as an independent module located before the second (string match) sieve. For the architecture of our classifier, we follow the approach of Muzny et al. (2017), although we replace the MaxEnt model with a feed-forward neural network. As adding neural classifiers to dutchcoref proved successful on the subtasks of mention detection, mention attributes and pronoun resolution (van Cranenburgh et al., 2021), we expect to see a similar improvement on the subtask of quote attribution.

3 Data and Material

For our experiments, we work with Dutch literary novels from both the RiddleCoref (van Cranenburgh, 2019) and the OpenBoek (van Cranenburgh and van Noord, 2022) corpora. For the task of quote attribution specifically, we needed to annotate the novels ourselves in order to obtain gold data. We will discuss both the corpora statistics and the annotation process below.

The RiddleCoref corpus was first presented in van Cranenburgh (2019) and consists of a selection Dutch (translated and original) contemporary literary novels from the *Riddle of Literary Quality* project (Koolen et al., 2020). This corpus contains a total of 33 documents, for which we use the train, development and test splits as defined in Poot and van Cranenburgh (2020). In total, there are 38,647 mentions in the corpus and on average 4,897.4 tokens per document. Unfortunately, the annotated texts from the RiddleCoref cannot be made publicly available due to copyright.

The OpenBoek corpus consists of public domain novels from Project Gutenberg enriched with several layers of annotation.[2] The corpus currently contains 9 fragments of Dutch-language novels and novellas. This corpus contains a total of 23,650 mentions, with an average of 11,502.4 tokens per document. The number of sentences per document (mean 643.3), as well as the number of tokens per document (> 10k), indicate that annotated OpenBoek fragments are longer than

[2] https://andreasvc.github.io/openboek/

the RiddleCoref fragments, or most other coreference datasets. These longer fragments lengths were chosen specifically with the challenge of long-document coreference resolution in mind.

4 Quote attribution annotation

Whereas gold standard coreference annotations (mentions and coreference clusters) were already available, this was not the case for the task of quote attribution. Therefore, we added quote attribution annotations as an extra annotation layer to the RiddleCoref and OpenBoek datasets. For the annotation we used the tool released by Muzny et al. (2017) along with our own annotation guidelines.[3] The guidelines can be summarized as follows:

1. We annotate all **direct** speech quotes, which often appear within quotation marks, or are preceded by a dash sign.
2. As for annotating mentions, the only mentions that should be annotated are the spans of text that refer to the speaker of a quote.
3. Each quote should be linked to the mention of that quote's speaker. A quote can only be linked to one mention, however one mention can be linked to multiple quotes.
4. In the case of multiple possible mention candidates for a quote's speaker, we will consistently choose the mention that is closest to the quote.
5. The mentions we annotate should always be outside the quotes they are connected to.

We only annotated the quotes and corresponding speaker mentions, but not the addressees for these quotes, as this is outside the scope of this paper.

Statistics In total, we annotated all 33 fragments from the RiddleCoref corpus and all 9 fragments from the OpenBoek corpus. Table 5 shows the number of quotes annotated per fragment. The RiddleCoref corpus contains a total of 1,864 quotes, whereas the OpenBoek corpus contains a total of 1,192 quotes. This results in an average of 56.5 quotes per fragment for the RiddleCoref corpus, versus an average of 132.4 quotes per OpenBoek fragment. The fact that the OpenBoek corpus seems to contain on average 2.3 times as much quotes per fragment is not surprising, as its fragments contain on average 2.1 times as many sentences per document. If we take this into account, the density of quotes per fragment is roughly the same for both corpora. We do however see that the number of quotes is more evenly distributed among the fragments of the OpenBoek corpora, whereas there seem to be more extreme outliers in the RiddleCoref corpus.

Inter-Annotator Agreement Ten fragments of 100 sentences from RiddleCoref had already been annotated at an earlier stage by the first author, allowing us to look at inter-annotator agreement with the annotations done for this project by the second author. Both annotators are native speakers of Dutch. For these 10 fragments of 100 words, we obtain an average F1-score of 83.7% (based on whether quotes are assigned to the correct speaker cluster, see Section 6). This is a lower bound, as the existing annotations were made before the annotation guidelines had been formalized. For more details and examples, see Section A.2.

5 Method

5.1 System architecture

We train a feed-forward classifier, using the aforementioned train and development split of the RiddleCoref corpus. This is a binary classifier that predicts for a given quote-mention candidate pair whether the mention is the speaker of the quote. Both the quotes and the candidate mentions are detected beforehand by the dutchcoref system, as we only focus on the attribution of each quote to the right speaker.

As candidate mentions, we only consider names, nouns and specific types of pronouns, that appear within a distance of at most one paragraph on either side of the quote. We restrict pronouns to personal and possessive pronouns, but unlike Muzny et al. (2017) we did not find restricting pronouns to only singular gendered pronouns to be helpful. Furthermore, mentions that appear within the quote are also excluded as its candidate mentions.

For each quote-mention pair, the classifier assigns a probability, which we use to select the most likely speaker for that quote. From all candidate mentions, we choose the mention with the highest probability. However, if this probability is lower than a pre-defined threshold (initially set at 0.2), no speaker is attributed to the quote.

Figure 1 provides an overview of our classifier. It consists of an input layer to which we apply a dropout of 0.2, followed by two dense hidden layers of 500 and 150 neurons, both with a dropout of 0.5. These layers both have ReLU activation

[3] https://github.com/frenkvdberg/dutchqa/blob/main/annotation_guidelines.pdf

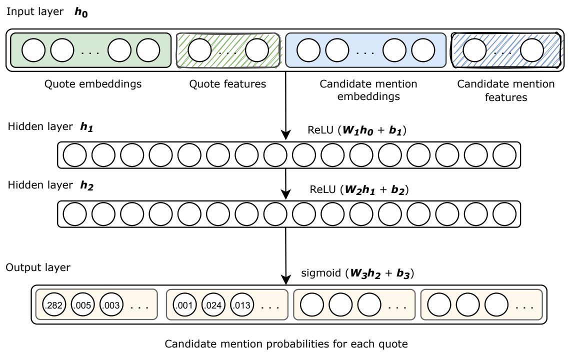

Figure 1: An overview of the classifier.

and batch normalization. For the output layer we use a sigmoid activation with L_2 regularization of 0.05. Furthermore, we use the Adam optimizer with a learning rate of 0.0001 and a batch size of 32. Lastly, we use early stopping to stop training when the model does not improve for 5 successive epochs.

Whereas van Cranenburgh (2019) favored precision over recall for their quote attribution component, we decide to experiment with tuning our classifier for different evaluation metrics. Using the same architecture with different features during training, we create a three variants of our classifier. Each focused on achieving the best performance on a specific metric, we present a *+precision*, *+recall* and *+F1* model. The features types are described in the next section; the classifier variants, along with their impact, are discussed in Section 6.1.

5.2 Features

Our classifier uses as input BERT embeddings and various handpicked features. For both the quotes and the mention candidates, we use BERT token embeddings produced by BERTje (de Vries et al., 2019), a pretrained, monolingual Dutch model. When a quote or a mention consists of multiple tokens, we take the mean of the embeddings of all tokens to use as input. As for the handpicked features, we will summarize each feature below.

Mention type of candidate mention; possible values: name, noun, pronoun.

Mention attributes For the mention attributes, we consider person, gender and animacy. The person attribute has three possible values: first, second, and third person. The gender attribute is either f (female), m (male), fm (mixed or unknown gender), or n (neuter). Lastly, animacy refers to whether the mention is human or non-human.

Quote length The number of tokens in the quote span.

Paragraph distance The number of paragraphs between the mention and the quote.

Token distance The number of tokens between the mention and the quote.

Quote distance The number of tokens between the end of the previous quote and the start of the current quote.

Mention occurrence in previous quote While we do not yet know the addressees for each quote, this feature might provide similar information. It looks at whether the candidate mention occurs within the previous quote, which means that it might be a speaker that is addressed before taking the next turn. Additionally, we store whether not the candidate mention itself, but a mention within the same cluster as the candidate mention has occurred within the previous quote.

Quotes in between The number of other quotes that appear between the current quote and the mention candidate.

Subject of speech verb Lastly, we check whether the candidate mention is the subject of a reported speech verb, for example ‘says’, ‘asks’ or ‘replies’. Such verbs are also referred to as cue words (Pareti, 2012, 2016; Papay and Padó, 2020). Note that the reported speech verbs are not part of the annotations, but are detected using a predefined list mined from a large corpus of parsed novels using a syntactic query of the form “NP verb quoted-speech” in various orders (van Cranenburgh, 2019).

5.3 Baseline methods

In order to gain a better insight into the performance of our classifier, we compare it to three different baselines. We will describe the approach of these baselines below in order of complexity.

Closest mention baseline Always choose the mention that is closest to the quote in terms of token distance. This closest mention is still chosen from the pool of candidate mentions, meaning that it is required to either be a name, noun, personal pronoun or a possessive pronoun. Inspired by Bamman et al. (2014) and Muzny et al. (2017).

Embeddings-only baseline A classifier as in Section 5.1, using only the BERT token embeddings for the quotes and the mentions in order to predict the speaker for each quote.

Dutchcoref baseline Since our goal is to improve the quote attribution performance of the rule-based dutchcoref system with a neural classifier, we need to know how well the rule-based approach (cf. Section 2.2) performs.

6 Evaluation

As mentioned before, for the RiddleCoref corpus we use the same train, development and test splits as defined in Poot and van Cranenburgh (2020). For the OpenBoek corpus however, there is no predefined split. A first proposal for this corpus was to use the novel *Max Havelaar* as development and *Eline Vere* as test, leaving the other seven novels as the train split. However, we noticed some poor performance with regards to the quote extraction part, which means these novels might not be representative as evaluation data. Especially for the fragment *Eline Vere*, which is the only fragment in which quotes are always introduced by a dash sign instead of quotation marks, the quotes were often extracted incorrectly. In the fragment *Max Havelaar*, the quotes are introduced by both quotation marks and dash signs in a very inconsistent manner. Moreover, quotes do not always have ending quotation marks.

Since quote extraction works well for the seven other fragments, we decided on the following: We will evaluate the performance of our classifiers, which were trained on the RiddleCoref train split, on both the RiddleCoref test split and on the seven remaining novels from the OpenBoek corpus (thus excluding *Eline Vere* and *Max Havelaar*). This way we can see whether the performance is better on a specific corpus, as well as analyze the potential differences.

We will report precision, recall and F1-scores, which were also used in earlier work (Muzny et al., 2017; van Cranenburgh, 2019). We report only scores indicating whether the quote was attributed to the correct speaker cluster. We do not report whether the quote was attributed to the same speaker mention as in the gold data, since this is a somewhat arbitrary annotation choice.

The evaluation can be further divided. During the training of our classifier and our initial experiments, we made use of gold standard coreference files that were already available for the RiddleCoref dataset. We will report the scores obtained by our classifier and the baseline systems when using these gold coreference files, meaning these systems have access to all the manually corrected mentions when making their predictions.

System	Threshold	P	R	F1
baselines:				
closest mention	N/A	40.7	40.7	40.7
embeddings only	0.20	53.9	53.6	53.7
dutchcoref	N/A	88.4	62.9	73.5
classifiers:				
neural +precision	0.20	91.5	58.8	71.6
neural +recall	0.20	85.8	67.9	75.8
neural +F1	0.20	87.6	67.9	76.5
classifiers w/ optimal thresholds:				
neural +precision	0.24	**92.9**	57.4	71.0
neural +recall	0.02	79.4	**77.2**	78.3
neural +F1	0.09	85.5	74.7	**79.8**

Table 1: Quote attribution on the RiddleCoref dev. set, using gold coreference, with classifiers optimizing precision, recall, or F1.

Additionally, we will implement the classifier into the existing dutchcoref system as an independent module. This way, we can compare its performance as a part of the dutchcoref system and compare whether it actually improves the rule-based approach in a realistic, end-to-end setting.

It must be noted that quotes that do not have a gold speaker are not taken into consideration during this evaluation, as this should be addressed as part of the quote extraction process, which is not the focus of this paper.

In the following subsections, we first report the results achieved in our experiments using gold standard coreference files. For these results, we also show which features contributed the most to each of our classifiers. Then, we report the results that our classifiers achieved when implemented into the dutchcoref system.

6.1 Results with gold coreference files

We first report the quote attribution results that were obtained when training the classifier on the RiddleCoref development set, using the available gold-standard coreference files. This way, we can see how well each system would perform in an ideal setting, where the quote attribution performance is not influenced by how well the dutchcoref system performs on other subtasks, as this evaluation setting will be discussed in Section 6.2.

Table 1 shows the performance of the baselines, as well as our neural classifiers, both with and without optimized thresholds. The simple baselines of always attributing a quote to the closest candidate mention or only using BERT embeddings as features are heavily outperformed by the rule-based dutchcoref module. However, we were able to train three different neural classifiers, each focused on outperforming the dutchcoref system on a specific evaluation metric.

We first apply the same probability threshold of 0.2 (below which no speaker will be assigned) to all classifiers in order to make an initial performance comparison. When looking at the speaker cluster scores, we see that the *+precision* classifier achieves an improvement of 3.1% on the precision metric over the dutchcoref system, although it performs worse in terms of recall and F1-score. Similarly, the *+recall* classifier outperforms the dutchcoref system by 5.0% on the recall metric, and the *+F1* classifier outperforms the dutchcoref system by 3.1% on the F1 metric. However, none of the classifiers outperforms the dutchcoref systems on all three metrics.

Then, we experimented with the probability thresholds in order to further improve the performance of our classifiers at their respective metrics. Increasing the threshold results in a higher precision score, while decreasing the thresholds results in a higher recall score. After optimizing these threshold values, the *+precision* classifier now achieves a precision score of 4.5% higher than the dutchcoref system. The *+recall* and *+F1* classifiers outperform the dutchcoref system on both recall and F1-score, with the *+recall* classifier achieving a recall score 14.3% higher and the *+F1* classifier achieving an F1-score 6.3% higher than the dutchcoref system.

In order to see which features contribute the most to each classifier's performance, we performed ablation experiments. Table 2 shows the performance of each classifier when removing one feature at a time. The *paragraph distance* feature seems to be by far the most important feature in all the three classifiers. Removing this feature would even mean that the *+precision* and *+F1* classifiers no longer outperform the dutchcoref system on their respective metrics. Interestingly, the *mention type* feature does not seem to contribute that much to the performance of each classifier. Removing this feature from the *+precision* classifier would result in slightly higher precision scores, however the F1-score would noticeably drop. Lastly, the *quote length* and *mention occurrence in previous quote* features that we introduced in Section 5.2 are not included in any of our three classifiers. While these features seemed to increase the scores in our initial experiments, they will unfortunately only decrease the performance when added to our final three classifiers.

Feature	P	R	F1
neural +precision:	92.9	57.4	71.0
- mention type	93.1	55.2	69.3
- mention attr. (excl. gender info)	88.6	55.5	68.2
- paragraph distance	**72.0**	50.8	59.6
- token distance	88.0	52.5	65.7
- quote distance	86.1	64.8	74.0
neural +recall:	79.4	77.2	78.3
- mention type	77.9	74.7	76.3
- mention attr. (excl. gender info)	76.6	74.7	75.7
- paragraph distance	69.8	**67.9**	68.8
- token distance	76.8	75.5	76.2
- subject of speech verb	76.8	74.7	75.8
neural +F1:	85.5	74.7	79.8
- mention type	84.1	72.5	77.9
- mention attributes	85.3	72.0	78.1
- paragraph distance	75.1	69.8	**72.4**
- token distance	83.9	73.1	78.1
- subject of speech verb	84.3	73.6	78.6
- quotes in between	81.3	73.1	77.0

Table 2: Ablation experiments for each classifier, removing one feature at a time.

QA module	**Set**	P	R	F1
rule-based	RC - dev	87.2	53.1	64.8
neural +precision	RC - dev	**94.6**	50.3	63.5
neural +recall	RC - dev	75.5	**71.4**	**73.3**
neural +F1	RC - dev	78.7	66.5	71.7
rule-based	RC - test	85.4	45.0	58.1
neural +precision	RC - test	**90.4**	43.4	58.2
neural +recall	RC - test	67.3	**65.0**	**66.1**
neural +F1	RC - test	72.9	58.8	64.7
rule-based	OpenBoek	**85.3**	64.0	72.8
neural +precision	OpenBoek	84.5	56.4	66.4
neural +recall	OpenBoek	76.0	**73.5**	**74.7**
neural +F1	OpenBoek	79.3	70.6	**74.7**

Table 3: End-to-end quote attribution results of different modules in the dutchcoref system.

6.2 Results with a coreference pipeline

For a more realistic evaluation of the performance of our classifiers, we compare the achieved scores again after implementing them as neural modules in the dutchcoref system. In Table 3, we report scores obtained on the RiddleCoref development and test sets, as well as on the selected seven OpenBoek novels. As expected, now that the quote attribution performance is dependent on the input received from earlier dutchcoref sieves, the scores achieved on the RiddleCoref development set are somewhat lower for all systems when compared to the scores from Table 1. Still, the neural classifiers each outperform the dutchcoref system on their respective metrics. It is interesting that this time, the *+recall* classifier obtains the highest recall **and** the highest F1-score of all four systems, both for the speaker mentions and for the speaker clusters.

For the RiddleCoref test set, all the scores are noticeably lower than they are for the development set. Again the *+recall* classifier achieves the the highest recall and F1-scores for the speaker clusters, although the *+F1* classifier does perform the best on the F1 metric if we look specifically at the speaker mentions performance. Seeing these relatively low scores on the test set inspired us to perform an error analysis, which we will discuss in Section 7.2.

Lastly, the quote attribution performance on the OpenBoek novels yields the highest F1-scores for all systems. This seems to be mostly due to all the recall scores being noticeably higher than they are for the RiddleCoref data splits. However, the *+precision* classifier does not outperform the rule-based approach on these seven novels. Furthermore, the rule-based approach achieves the highest F1 score looking purely at the speaker mentions and for the speaker clusters the difference in F1 scores between the rule-based approach and the best performing neural classifiers is noticeably smaller than on the RiddleCoref novels. For transparency, we included the results on each individual novel in Appendix A.3.

7 Analysis

To gain a better understanding of the challenges of literary quote attribution, we now take a closer look at the test data and model outputs. We first consider the distribution of quote types, and then perform an error analysis of the systems we evaluated.

7.1 Quote type distribution

In order to compare the RiddleCoref test novels to the OpenBoek novels, we consider the distribution of the quote types per novel (see Section A.4 for detailed statistics). As mentioned before, we distinguish between four different quote types: **explicit** (said Tom), **anaphoric pronoun** (said he), **anaphoric other** (said his friend) and **implicit**. Looking at the relative frequencies, we see that the percentage of anaphoric other quotes is roughly the same for both datasets. However, we see a big difference in the relative amount of implicit quotes: 57% for the RiddleCoref test novels vs only 32% for the OpenBoek novels. Whereas implicit quotes are by far the most prominent in the RiddleCoref test novels, anaphoric pronoun quotes are the most prominent in the OpenBoek novels, slightly surpassing the implicit quotes. The large dataset of classic English novels by Vishnubhotla et al. (2022) has about 36% implicit and 29% anaphoric quotes (based on Table 5). This figure is similar to that of OpenBoek and suggests that contemporary novels may contain more implicit quotes than classic novels, which makes the task of quote attribution harder for contemporary novels.

Furthermore, it is interesting to see that even within the datasets the quote type distribution can differ considerably per novel. For instance, in the novel *Gooische Vrouwen*, 67% of the quotes are explicit, whereas this percentage is only 4% for *Cobra* and 0% for *Mannentester*. Similar outliers can be seen for the OpenBoek novels, where *De Agra Schat* contains 61% quotes of type anaphoric pronoun, but *Reis Om De Wereld* contains only 12% quotes of the same type. Some of this variance could be attributed to genre, but also to author style.

This distribution helps us better understand the difference in performance of our classifiers on these datasets, which we discuss in the next subsection.

7.2 Error analysis

Looking at the speaker cluster F1-scores in Table 3, we see a large difference in performance on the RiddleCoref test novels and the OpenBoek novels. This difference is not only visible for our neural classifiers, but also for the rule-based approach, which achieved 14.7% F1 points higher on the OpenBoek novels. As the performance was the worst on the RiddleCoref test novels, we analyzed the mistakes that the different systems made on each novel. See Section A.4 for a breakdown of mistakes per quote type and novel.

We see that for each system the majority of the mistakes are made on implicit quotes. Even by our best classifier, these quotes are still incorrectly classified in 53% of the cases, with our worst performing classifier incorrectly classifying these quotes 76% of the time. The anaphoric pronoun quotes seem to be the easiest to classify for each system, especially for the *+recall* and *+F1* classifiers, which only make mistakes on 2% of these quotes.

Looking at these mistakes in combination with the quote distribution of Table 10 helps us understand the difference in performance on the aforementioned datasets. As can be seen from the quote distribution, the RiddleCoref test novels contain 57% implicit and 20% anaphoric pronoun quotes, whereas these percentages are 32% implicit and 35% anaphoric pronoun quotes for the OpenBoek novels. Seeing how by far the most mistakes are made on implicit quotes, it is only natural to see a worse performance on a dataset with novels that contain on average more of these implicit quotes.

It is also interesting to see how our neural *F1* classifier substantially outperforms the rule-based approach for explicit-, anaphoric other- and especially anaphoric pronoun quotes, but only slightly for the implicit quotes. The *+precision* classifier actually performs worse than the rule-based approach only for the implicit quotes. This shows us that even with the features we presented and implemented in our classifiers, we still have an especially hard time attributing implicit quotes to the right speaker.

Lastly, we see that for each system, anaphoric non-pronoun quotes are substantially harder to classify than anaphoric pronoun quotes, which is in line with the results presented in (Muzny et al., 2017).

8 Conclusion

In this paper, we focused on training a classifier to improve the task of quote attribution when compared to dutchcoref's rule-based approach. We trained three different feed-forward neural network classifiers, each one focused a different metric for speaker clusters: precision, recall and F1-score. For the task of quote attribution, we manage to improve on the rule-based approach by 8.0% F1 points on the RiddleCoref test novels and by 1.9% F1 points on the OpenBoek novels.

With our quote attribution error analysis we show that each system makes the most mistakes on implicit quotes. Moreover, anaphoric pronoun quotes prove to be harder than anaphoric non-pronoun quotes, as each of the systems performs the best on the anaphoric pronoun quote type. This also explains why the quote attribution performance on the OpenBoek novels is notably higher than on the RiddleCoref test novels, as the OpenBoek novels contain relatively more anaphoric pronoun quotes and less implicit quotes.

For future work, we think there is still a lot of improvement to be gained, especially on implicit quotes. As we have found features that reduce mistakes on anaphoric pronoun quotes by 91.7% with respect to the rule-based approach, future experiments can look specifically at how to decrease mistakes on implicit quotes. Furthermore, we did not consider the task of identifying addressees, thus having to rely on the rule-based approach to identify these after we first identify the speakers using our neural classifiers. Jointly identifying speakers and addressees may yield additional performance gains, since it would enable the classifier to pick up on turn-taking patterns in a data-driven manner. Reported speech verbs (also known as cue words) were not part of the annotations, but detected using a predefined list. Recall may be improved by detecting them using a classifier trained on annotated cue words. In terms of machine learning, fine-tuning BERT for the task of quote attribution (rather than simply using averaged token embeddings as features) and/or incorporating more context with for example an LSTM on top of the BERT embeddings can be expected to yield additional improvements.

Lastly, we think further improvements can also be made on quote extraction, as we saw that there were still a lot of mistakes made on the OpenBoek novels *Max Havelaar* and *Eline Vere*. As most of these mistakes were made on quotes starting with a dash sign, more elaborate rules targeting these kind of quotes could improve the overall performance of dutchcoref even more.

Acknowledgements

The authors are grateful to Khalid Al-Khatib and three anonymous reviewers for helpful comments.

References

Mariana S. C. Almeida, Miguel B. Almeida, and André F. T. Martins. 2014. A joint model for quotation attribution and coreference resolution. In *Proceedings of EACL*, pages 39–48.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of ACL*, pages 370–379.

Julian Brooke, Adam Hammond, and Graeme Hirst. 2017. Using models of lexical style to quantify free indirect discourse in modernist fiction. *Digital Scholarship in the Humanities*, 32(2):234–250.

Joanna Byszuk, Michał Woźniak, Mike Kestemont, Albert Leśniak, Wojciech Łukasik, Artjoms Šeļa, and Maciej Eder. 2020. Detecting direct speech in multilingual collection of 19th-century novels. In *Proceedings of LT4HALA*, pages 100–104.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. arXiv:1912.09582.

David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of ACL*, pages 138–147.

David K. Elson and Kathleen R. McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Kevin Glass and Shaun Bangay. 2007. A naive salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA'07)*, pages 1–6.

Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. In *Proceedings of ACL*, pages 1312–1320.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of EACL*, pages 427–431.

Corina Koolen, Karina van Dalen-Oskam, Andreas van Cranenburgh, and Erica Nagelhout. 2020. Literary quality in the eye of the Dutch reader: The national reader survey. *Poetics*.

Eve Kraicer and Andrew Piper. 2019. Social characters: The hierarchy of gender in contemporary English-language fiction. *Journal of Cultural Analytics*, 3(2).

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys*, 52(5).

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):885–916.

Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. 2017. A two-stage sieve approach for quote attribution. In *Proceedings of EACL*, pages 460–470.

Tim O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of EMNLP-CoNLL*, pages 790–799.

Tim O'Keefe, Kellie Webster, James R. Curran, and Irena Koprinska. 2013. Examining the impact of coreference resolution on quote attribution. In *Proceedings of ALTA*, pages 43–52.

Sean Papay and Sebastian Padó. 2020. RiQuA: A corpus of rich quotation annotation for English literary text. In *Proceedings of LREC*, pages 835–841.

Silvia Pareti. 2012. A database of attribution relations. In *Proceedings of LREC*, pages 3213–3217.

Silvia Pareti. 2016. PARC 3.0: A corpus of attribution relations. In *Proceedings of LREC*, pages 3914–3920.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of EMNLP*, pages 989–999.

Corbèn Poot and Andreas van Cranenburgh. 2020. A benchmark of rule-based and neural coreference resolution in Dutch novels and news. In *Proceedings of CRAC*, pages 79–90.

Andrew Salway, Paul Meurer, Knut Hofland, and Øystein Reigem. 2017. Quote extraction and attribution from norwegian newspapers. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 293–297.

Elena Semino and Mick Short. 2004. *Corpus Stylistics: Speech, writing and thought presentation in a corpus of English writing*. Routledge Advances In Corpus Linguistics. Routledge, London.

Matthew Sims and David Bamman. 2020. Measuring information propagation in literary social networks. In *Proceedings of EMNLP*, pages 642–652.

Ted Underwood, David Bamman, and Sabrina Lee. 2018. The transformation of gender in english-language fiction. *Journal of Cultural Analytics*, 3(2).

Andreas van Cranenburgh. 2019. A Dutch coreference resolution system with an evaluation on literary fiction. *Computational Linguistics in the Netherlands Journal*, 9:27–54.

Andreas van Cranenburgh, Esther Ploeger, Frank van den Berg, and Remi Thüss. 2021. A hybrid rule-based and neural coreference resolution system with an evaluation on Dutch literature. In *Proceedings of CRAC*, pages 47–56.

Andreas van Cranenburgh and Gertjan van Noord. 2022. Openboek: A corpus of literary coreference and entities with an exploration of historical spelling normalization. *Computational Linguistics in the Netherlands Journal*, 12:235–251.

Krishnapriya Vishnubhotla, Adam Hammond, and Graeme Hirst. 2022. The project dialogism novel corpus: A dataset for quotation attribution in literary texts. In *Proceedings of LREC*, pages 5838–5848.

Chak Yan Yeung and John Lee. 2017. Identifying speakers and listeners of quoted speech in literary works. In *Proceedings of IJCNLP*, pages 325–329.

Michael Yoder, Sopan Khosla, Qinlan Shen, Aakanksha Naik, Huiming Jin, Hariharan Muralidharan, and Carolyn Rosé. 2021. FanfictionNLP: A text processing pipeline for fanfiction. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 13–23.

	RiddleCoref			OpenBoek
	train	dev	test	
documents	23	5	5	9
sentences	6,803	1,525	1,536	5,709
sentences per document	295.8	305.0	307.2	643.3
average sentence length	15.5	18.4	18.3	18.1
tokens	105,517	28,042	28,054	103,522
mentions	25,194	6,584	6,869	23,650
entities	9,041	2,643	3,008	8,875
mentions / entities	2.79	2.49	2.28	2.66
mentions / tokens	0.24	0.23	0.24	0.23
entities / tokens	0.09	0.09	0.11	0.09
% pronouns	40.4	35.7	38.1	40.9
% nominals	47.0	49.4	52.8	48.0
% names	12.6	14.9	9.1	11.1

Table 4: RiddleCoref and OpenBoek corpora statistics.

A Appendices

A.1 Corpus statistics

See Table 4 for various statistics of the two corpora used in this paper. Table 5 lists the number of annotated quotes for each text (fragment).

A.2 Inter-Annotator Agreement

For the RiddleCoref corpus, ten of the fragments were annotated by the first author at an earlier stage, allowing us to look at inter-annotator agreement. However, these annotations were made before the annotation guidelines were created, so some inconsistencies are to be expected. While this score is often calculated using Cohen's kappa, we found that this method was not applicable here, as there is not always a clear number of possible speaker mentions and speaker clusters to which a quote could be attributed. Instead, we evaluated our annotations against the other annotations to calculate the F1-scores for each fragment, which can be found in Table 6.

Looking at the F1-scores for the clusters, we see that we achieve an average F1-score of 83.7, indicating that we often attribute quotes to the same speaker cluster. This score is noticeably lower for the mentions, as the choice of which mention to pick is a rather arbitrary choice, which is best shown for the novel by Barnes. For the 23 quotes this novel contains, we tend to attribute the quotes to different mentions, achieving an F1-score of 9.3, while still often attributing these quotes to the same speaker cluster, as can be seen from the cluster F1-score of 73.4. Ultimately, it is the speaker cluster that should be correctly recognized by our final system, so getting these correct is what matters most.

Still, for the speaker cluster scores we see two especially low scores. For the novel *De begraafplaats van Praag*, we see an F1-score of 40.0. Here, the challenge is whether the detected quotes are meant to be attributed to a speaker or not, as we show for the following two detected quotes:

(5) *(...) je hoeft alleen maar af te geven op een ander volk, dus bijvoorbeeld ["wij Polen hebben dat en dat manco"]*$_{\textit{quote}}$*, of ze zeggen meteen, omdat ze voor niemand onder willen doen, zelfs niet als het iets negatiefs betreft: ["O nee, hoor! Hier in Frankrijk zijn we veel erger"]*$_{\textit{quote}}$*, waarna ze aan een anti-Franse tirade beginnen die pas eindigt als het tot ze doordringt dat ze erin zijn getuind.*
(...) you only have to speak ill of another people, for example ["we Poles have such and such a defect"]$_{\text{quote}}$, and since they do not want to be second to anyone, even in wrong, they react with: ["Oh no, here in France we are worse"]$_{\text{quote}}$, and they start running down the French until they realize they've been caught out.

Whereas the other annotator does not assign a speaker to either of these quotes, we argue that the second quote can be attributed to the underlined mention *ze* (they). This scenario is repeated for another quote in the novel, where the other annotator

Riddlecoref - train	# quotes
Abdolah, Koning	94
Barnes, Alsof Voorbij Is	23
Bernlef, Zijn Dood	104
Bezaz, Vinexvrouwen	11
Binet, Hhhh	23
Carre, Ons Soort Verrader	9
Collins, Hongerspelen	129
Dewulf, Kleine Dagen	11
Eco, Begraafplaats Van Praag	3
Eggers, Wat Is Wat	19
Grunberg, Huid En Haar	19
James, Vijftig Tinten Grijs	11
Kinsella, Shopaholic Baby	51
Koch, Diner	12
Mansell, Versier Me Dan	41
Moor, Schilder En Meisje	5
Rowling, Harry Potter	468
Siebelink, Oscar	57
Vermeer, Cruise	23
Voskuil, Buurman	54
Weisberger, Chanel Chic	7
Worthy, James Worthy	15
Yalom, Raadsel Spinoza	20
Total	***1,209***

Riddlecoref - dev	# quotes
Gilbert, Eten Bidden Beminnen	9
Kluun, Haantjes	16
Kooten, Verrekijker	57
Mitchell, Niet Verhoorde Gebeden	222
Springer, Quadriga	82
Total	***386***

Riddlecoref - test	
Forsyth, Cobra	46
Japin, Vaslav	114
Proper, Gooische Vrouwen	36
Royen, Mannentester	25
Verhulst, Laatste Liefde Van	48
Total	***269***

OpenBoek	
Conan Doyle, De Agra Schat	186
Couperus, Eline Vere	101
Hugo, De Ellendigen	78
Multatuli, Max Havelaar	31
Nescio, De Uitvreter	220
Nescio, Dichtertje	150
Nescio, Titaantjes	91
Tolstoy, Anna Karenina	182
Verne, Reis Om De Wereld	153
Total	***1,192***

Table 5: Number of quotes per text.

again does not assign a speaker, while we do. As the fragment of this novel contains very few quotes, each difference in our annotations heavily lowers the inter-annotator agreement score.

For the novel *Het diner*, the F1-score of 59.5 can also be explained by us assigning speakers to quotes more often than the other annotator does, as we show in example (6):

(6) *Maar ik noem haar zelden mijn vrouw — bij officiële gelegenheden af en toe, in zinnen als: ['Mijn vrouw kan op dit moment niet aan de telefoon komen']$_{quote}$, of: ['Mijn vrouw weet toch echt zeker dat zij een kamer met uitzicht op zee had gereserveerd.']$_{quote}$*
But I rarely refer to her as my wife — on official occasions sometimes, in sentences like ['My wife can't come to the phone right now']$_{\text{quote}}$, or: ['My wife is very sure she asked for a room with a sea view.']$_{\text{quote}}$

Again, both quotes are not assigned a speaker by the other annotator, whereas we attribute both quotes to the underlined mention *ik* (I). We notice that the quotes on which we disagree are often introduced by phrases like *bijvoorbeeld* (for example) or *zoals* (such as). These quotes can sometimes be interpreted as describing hypothetical dialogue, leaving the reader uncertain whether the dialogue has actually ever taken place. Still, we choose to assign these examples of dialogue to the intended speaker, causing our annotations to differ with the other annotator at times.

A.3 Quote attribution performance per novel

- RiddleCoref development set: cf. Table 7.
- RiddleCoref test set: cf. Table 8.
- OpenBoek novels: cf. Table 9.

Novel	Mentions F1	Clusters F1
Barnes, Alsof het voorbij is	9.3	73.4
Carre, Ons soort verrader	53.7	100
Eco, Begraafplaats van Praag	40.0	40.0
Eggers, Wat is de wat	52.6	100
Grunberg, Huid en haar	85.7	100
James, Vijftig tinten grijs	34.1	100
Koch, Diner	61.1	59.5
Moor, De schilder en het meisje	100	100
Voskuil, De buurman	76.3	97.7
Yalom, Het raadsel Spinoza	62.5	66.7
Average	***57.5***	***83.7***

Table 6: Annotator agreement on 10 RiddleCoref texts.

A.4 Analysis: detailed tables

Table 10 lists the number of quote types per annotated text. Table 11 lists the number of mistakes broken down by quote type in the texts of the RiddleCoref test set.

Novel	**Mentions**			**Clusters**		
	P	R	F1	P	R	F1
rule-based:						
Gilbert, Eten Bidden Beminnen	100	44.4	61.5	100	44.4	61.5
Kluun, Haantjes	85.7	37.5	52.2	85.7	37.5	52.2
Kooten, Verrekijker	78.6	64.7	71.0	81.0	66.7	73.1
Mitchell, Niet Verhoorde Gebeden	84.2	67.3	74.8	88.9	71.0	79.0
Springer, Quadriga	71.7	40.7	52.0	80.4	45.7	58.3
Average	***84.0***	***50.9***	***62.3***	***87.2***	***53.1***	***64.8***
neural +precision:						
Gilbert, Eten Bidden Beminnen	100	44.4	61.5	100	44.4	61.5
Kluun, Haantjes	100	25.0	40.0	100	25.0	40.0
Kooten, Verrekijker	90.7	76.5	83.0	95.3	80.4	87.2
Mitchell, Niet Verhoorde Gebeden	89.9	57.9	70.5	92.8	59.8	72.7
Springer, Quadriga	82.5	40.7	54.5	85.0	42.0	56.2
Average	***92.6***	***48.9***	***61.9***	***94.6***	***50.3***	***63.5***
neural +recall:						
Gilbert, Eten Bidden Beminnen	88.9	88.9	88.9	100	100	100
Kluun, Haantjes	53.8	43.8	48.3	61.5	50.0	55.2
Kooten, Verrekijker	84.3	84.3	84.3	88.2	88.2	88.2
Mitchell, Niet Verhoorde Gebeden	69.2	68.2	68.7	74.4	73.4	73.9
Springer, Quadriga	63.6	60.5	62.0	77.9	74.1	75.9
Average	***72.0***	***64.2***	***65.8***	***75.5***	***71.4***	***73.3***
neural +F1:						
Gilbert, Eten Bidden Beminnen	88.9	88.9	88.9	100	100	100
Kluun, Haantjes	50.0	31.2	38.5	60.0	37.5	46.2
Kooten, Verrekijker	86.0	84.3	85.1	90.0	88.2	89.1
Mitchell, Niet Verhoorde Gebeden	78.6	68.7	73.3	84.0	73.4	78.3
Springer, Quadriga	68.7	56.8	62.2	80.6	66.7	73.0
Average	***70.8***	***60.2***	***64.8***	***78.7***	***66.5***	***71.7***

Table 7: Quote attribution scores per novel on the RiddleCoref development set, when classifiers are implemented within the dutchcoref system.

Novel	**Mentions**			**Clusters**		
	P	R	F1	P	R	F1
rule-based:						
Forsyth, Cobra	84.2	34.8	49.2	84.2	34.8	49.2
Japin, Vaslav	90.3	57.5	70.3	95.8	61.1	74.6
Proper, Gooische Vrouwen	76.9	57.1	65.6	80.8	60.0	68.9
Royen, Mannentester	81.8	36.0	50.0	81.8	36.0	50.0
Verhulst, Laatste Liefde Van	78.9	31.2	44.8	84.2	33.3	47.8
Average	***82.4***	***42.3***	***56.0***	***85.4***	***45.0***	***58.1***
neural +precision:						
Forsyth, Cobra	85.0	37.0	51.5	85.0	37.0	51.5
Japin, Vaslav	87.3	42.5	57.1	89.1	43.4	58.3
Proper, Gooische Vrouwen	87.0	57.1	69.0	87.0	57.1	69.0
Royen, Mannentester	81.8	36.0	50.0	90.9	40.0	55.6
Verhulst, Laatste Liefde Van	100	39.6	56.7	100	39.6	56.7
Average	***88.2***	***42.4***	***56.9***	***90.4***	***43.4***	***58.2***
neural +recall:						
Forsyth, Cobra	57.6	41.3	48.1	57.6	41.3	48.1
Japin, Vaslav	64.3	63.7	64.0	73.2	72.6	72.9
Proper, Gooische Vrouwen	62.9	62.9	62.9	62.9	62.9	62.9
Royen, Mannentester	52.0	52.0	52.0	64.0	64.0	64.0
Verhulst, Laatste Liefde Van	59.5	52.1	55.6	69.0	60.4	64.4
Average	***59.3***	***57.7***	***58.6***	***67.3***	***65.0***	***66.1***
neural +F1:						
Forsyth, Cobra	67.9	41.3	51.4	71.4	43.5	54.1
Japin, Vaslav	70.4	61.1	65.4	81.6	70.8	75.8
Proper, Gooische Vrouwen	69.7	65.7	67.6	69.7	65.7	67.6
Royen, Mannentester	56.5	52.0	54.2	65.2	60.0	62.5
Verhulst, Laatste Liefde Van	70.6	50.0	58.5	76.5	54.2	63.4
Average	***67.0***	***54.0***	***59.4***	***72.9***	***58.8***	***64.7***

Table 8: Quote attribution scores per novel on the RiddleCoref test set, when classifiers are implemented within the dutchcoref system.

Novel	Mentions			Clusters		
	P	R	F1	P	R	F1
rule-based:						
Conan Doyle, De Agra Schat	92.8	76.2	83.7	94.7	77.8	85.5
Hugo, De Ellendigen	82.7	56.6	67.2	86.5	59.2	70.3
Nescio, De Uitvreter	89.5	70.0	78.6	91.4	71.5	80.2
Nescio, Dichtertje	66.2	35.3	46.1	75.0	40.0	52.2
Nescio, Titaantjes	71.4	49.5	58.4	77.8	53.8	63.6
Tolstoy, Anna Karenina	77.1	60.0	67.5	81.4	63.3	71.3
Verne, Reis Om De Wereld	86.2	78.3	82.1	90.6	82.2	86.2
Average	***80.8***	***60.8***	***69.1***	***85.3***	***64.0***	***72.8***
neural +precision:						
Conan Doyle, De Agra Schat	95.1	73.0	82.6	95.8	73.5	83.2
Hugo, De Ellendigen	80.7	60.5	69.2	86.0	64.5	73.7
Nescio, De Uitvreter	80.4	69.6	74.6	83.8	72.5	77.7
Nescio, Dichtertje	69.0	32.7	44.3	77.5	36.7	49.8
Nescio, Titaantjes	77.1	29.7	42.9	85.7	33.0	47.6
Tolstoy, Anna Karenina	81.5	61.1	69.8	85.9	64.4	73.7
Verne, Reis Om De Wereld	86.2	65.8	74.6	87.9	67.1	76.1
Average	***81.4***	***53.2***	***62.6***	***84.5***	***56.4***	***66.4***
neural +recall:						
Conan Doyle, De Agra Schat	77.7	75.1	76.4	79.3	76.8	78.0
Hugo, De Ellendigen	71.2	68.4	69.8	75.3	72.4	73.8
Nescio, De Uitvreter	76.3	76.3	76.3	84.5	84.5	84.5
Nescio, Dichtertje	53.5	50.7	52.1	66.9	63.3	65.1
Nescio, Titaantjes	50.6	49.5	50.0	67.4	65.9	66.7
Tolstoy, Anna Karenina	69.4	66.7	68.0	78.6	75.6	77.1
Verne, Reis Om De Wereld	75.7	71.7	73.6	79.9	75.7	77.7
Average	***67.8***	***65.5***	***66.6***	***76.0***	***73.5***	***74.7***
neural +F1:						
Conan Doyle, De Agra Schat	89.4	77.3	82.9	90.0	77.8	83.5
Hugo, De Ellendigen	80.0	68.4	73.8	83.1	71.1	76.6
Nescio, De Uitvreter	81.4	78.3	79.8	86.4	83.1	84.7
Nescio, Dichtertje	57.5	48.7	52.7	70.9	60.0	65.0
Nescio, Titaantjes	56.4	48.4	52.1	69.2	59.3	63.9
Tolstoy, Anna Karenina	70.8	66.1	68.4	79.2	73.9	76.4
Verne, Reis Om De Wereld	83.5	73.0	77.9	87.2	76.3	81.4
Average	***74.1***	***63.8***	***67.5***	***79.3***	***70.6***	***74.7***

Table 9: Quote attribution scores per novel on the selected OpenBoek novels, when classifiers are implemented within the dutchcoref system.

Novel	EXP	ANA-P	ANA-O	IMP
RiddleCoref - test:				
Forsyth, Cobra	2	5	9	30
Japin, Vaslav	5	39	1	69
Proper, Gooische Vrouwen	24	0	1	11
Royen, Mannentester	0	5	2	18
Verhulst, Laatste Liefde	12	4	7	25
Total	***43***	***53***	***20***	***153***
Relative total	***16%***	***20%***	***7%***	***57%***
OpenBoek:				
Conan Doyle, De Agra Schat	18	114	6	48
Hugo, De Ellendigen	8	31	22	17
Nescio, De Uitvreter	95	73	4	48
Nescio, Dichtertje	11	44	18	77
Nescio, Titaantjes	20	21	6	44
Tolstoy, Anna Karenina	43	70	12	57
Verne, Reis om de wereld	72	18	16	47
Total	***267***	***371***	***84***	***338***
Relative total	***25%***	***35%***	***8%***	***32%***

Table 10: Distribution of quote types in RiddleCoref test and OpenBoek texts. EXP: explicit; ANA-P: anaphoric pronoun; ANA-O: anaphoric other; IMP: implicit.

System	Novel	EXP	ANA-P	ANA-O	IMP
dutchcoref	Forsyth, Cobra	0	1	3	25
	Japin, Vaslav	1	6	0	37
	Proper, Gooische Vrouwen	7	0	0	7
	Royen, Mannentester	0	2	2	12
	Verhulst, Laatste Liefde Van	3	3	6	20
	Total	***11***	***12***	***11***	***101***
	mistakes / quotes	***0.26***	***0.23***	***0.55***	***0.66***
neural +precision	Forsyth, Cobra	0	1	3	25
	Japin, Vaslav	2	4	0	58
	Proper, Gooische Vrouwen	5	0	1	9
	Royen, Mannentester	0	1	1	13
	Verhulst, Laatste Liefde Van	0	2	5	22
	Total	***7***	***8***	***10***	***117***
	mistakes / quotes	***0.16***	***0.15***	***0.50***	***0.76***
neural +recall	Forsyth, Cobra	0	1	2	22
	Japin, Vaslav	2	0	0	28
	Proper, Gooische Vrouwen	5	0	1	7
	Royen, Mannentester	0	0	1	8
	Verhulst, Laatste Liefde Van	0	0	3	16
	Total	***7***	***1***	***7***	***81***
	mistakes / quotes	***0.16***	***0.02***	***0.35***	***0.53***
neural +F1	Forsyth, Cobra	0	0	2	23
	Japin, Vaslav	1	0	0	32
	Proper, Gooische Vrouwen	5	0	0	7
	Royen, Mannentester	0	1	1	8
	Verhulst, Laatste Liefde Van	0	0	3	19
	Total	***6***	***1***	***5***	***81***
	mistakes / quotes	***0.14***	***0.02***	***0.25***	***0.53***

Table 11: Mistakes per quote type on the RiddleCoref test novels.