


A Novel Approach for Identification and Linking of Short Quotations in Scholarly Texts and Literary Works

Frederik Arnold¹ 
Robert Jäschke¹ 

1. Berlin School for Library and Information Science, Humboldt-Universität zu Berlin , Berlin, Germany.

Citation

Frederik Arnold and Robert Jäschke (2023). "A Novel Approach for Identification and Linking of Short Quotations in Scholarly Texts and Literary Works". In: *Journal of Computational Literary Studies* 2 (1). [10.48694/jcls.3590](https://doi.org/10.48694/jcls.3590)

Date published 2024-01-30

Date accepted 2023-10-27

Date received 2023-01-31

Keywords

quotation linking, literary works, scholarly works, machine learning, language models

License

CC BY 4.0 

Reviewers

Artjoms Šeļa, Ryan Cordell

Note

This paper has passed through the conference track of JCLS. In addition to being peer reviewed, it was presented and discussed at the 2nd Annual Conference of Computational Literary Studies at Würzburg University in June 2023.

Abstract. We present two approaches for the identification and linking of short quotations between scholarly works and literary works: *ProQuo*, a specialized pipeline, and *ProQuoLM*, a more general language model based approach. Our evaluation shows that both approaches outperform a strong baseline and the overall performance is on the same level. We compare the performance of ProQuoLM on texts with and without (page) reference information and find that reference information is not used. Based on our findings, we propose the following steps for future improvements: Further analysis of the influence of a bigger context window for better handling of long distance references and the introduction of positional information of the literary work so that reference information can be (better) utilized.

1. Introduction

Scholarly and literary texts do not exist in a vacuum but rather interact in various ways: Literary scholars quote literary works, scholarly works and other sources, to support the reasoning of their interpretations or to build on earlier publications. Although (literary) interpretations usually only concretely refer to certain passages of a (literary) text, they often claim to interpret the entire work. We know little about the inner workings of this skillful selection, the corresponding attentional behavior, as well as the canonization of passages, that, for various reasons, lend themselves to support the interpretation. The intertextual relationships between interpretations and objects of interpretation vary in nature, ranging from relatively vague references to renderings of clearly identifiable passages of text in the interpreter's own words to direct quotations.

Long quotations, that is, quotations of a length of five words or more, can be identified using text reuse detection methods (Arnold and Jäschke 2021). Shorter quotations are a major challenge for reasons we will explain in a moment. They are important, however, either because they apply to particularly weighty words or because they are indicative of references to passages. Other uses include intertextuality research, for example, in the analysis of quotations from Hamlet (Hohl Trillini and Quassdorf 2010) or Shakespeare in general (Molz 2020), argument mining in scholarly texts where the context in the literary work is relevant to understand how texts are analyzed (Descher and Petraschka

Literary work	Scholarly work
<p>Wo ist die Hand so zart, daß ohne Irren Sie sondern mag beschränkten Hirnes Wirren, So fest, daß ohne Zittern sie den Stein Mag schleudern auf ein arm verkümmert Sein? Wer wagt es, eitlen Blutes Drang zu messen, Zu wägen jedes Wort, das unvergessen In junge Brust die zähen Wurzeln trieb, Des Vorurteils geheimen Seelendieb? Du Glücklicher, geboren und gehegt Im lichten Raum, von frommer Hand gepflegt, Leg hin die Waagschal, nimmer dir erlaubt! Laß ruhn den Stein – er trifft dein eignes Haupt!</p>	<p>Hier laufen wir Gefahr, uns zu jenem selbstbezogenen Messen und Wägen verführen zu lassen, das uns "Glücklichen", die "geboren und gehegt/Im lichten Raum, von frommer Hand gepflegt" (882) wurden, nicht erlaubt ist.</p>

Figure 1: Example shows an excerpt of a scholarly work (Schaum 2004) which quotes from a literary work (excerpt from Droste-Hülshoff 1979). A single word quotation is shown in green, a long quote in dark blue and a (page) reference in light blue.

2018, Winko and Jannidis 2015), or the identification of key passages, that is, passages that are particularly important to expert readers (Arnold and Fiechter 2022).

As already mentioned, quotations can be of varying length from single words to whole paragraphs. Bibliographic references, often in footnotes or a dedicated reference section, identify the work a quotation is taken from. Page references, either in footnotes or in parentheses in the running text, are often used to indicate specific pages.¹ Despite this information, identifying the exact source location of a quotation is a hard task.

Existing tools for the identification of quotations, for example, Copyfind (Bloomfield 2016), Passim (Smith et al. 2014), TextMatcher (Reeve 2020) or Quid (Arnold and Jäschke 2021), are not suitable for unambiguously identifying instances which are shorter than at least a couple of words, as they often rely on text reuse detection methods. For these shorter quotations, especially for quotations consisting of just one word, a number of challenges arise which the tools just mentioned cannot solve. Firstly, short quotations are much more likely to have multiple possible sources in the literary work, which makes it more difficult to link a quotation to its source. Secondly, quotations from other sources, for example, other scholarly works or quotations from the Bible, are much more likely to also occur in the literary work just by chance.

In this paper, we present and compare two tools for the identification and linking of short quotations between scholarly works and literary works: ProQuo and ProQuoLM. Quotations, long and short, are often accompanied by citation information, for example, page or line numbers, either in the running text in parentheses or in footnotes (Figure 1). Our main idea behind ProQuo is to use the references corresponding to long quotations as examples to distinguish references corresponding to short quotations from other text in parentheses and other references, for example, Bible references or references to other literary works. We then extract relations between short quotations and references and use that information and the position of long quotations as anchors to link short quotations to the literary work.

1. In this work, whenever we talk about *references*, we refer to the second type of reference, the one used to indicate specific pages.

We compare this specialized pipeline with its explainable steps to a more general, state-of-the-art neural language model approach which we named ProQuoLM. For this second approach, we first extract candidates for short quotations and then use a fine-tuned language model to filter the candidates. The comparison allows us to investigate and illustrate the advantages and disadvantages of a pipeline with explainable steps and a blackbox neural language model approach. This is especially relevant in light of recent discussions about computational approaches in digital humanities (Da 2019).

This paper is organized as follows: In [section 2](#), we provide an overview on related work. In [section 3](#) we describe our approaches. In [section 4](#) we present our dataset and experimental setup, followed by [section 5](#) where we present the results.

2. Related Work

Our task is related to reference extraction and segmentation, quotation detection and quotation attribution.

Existing tools for reference extraction and segmentation (*GROBID 2008–2022*; Prasad et al. 2018) focus on STEM fields (science, technology, engineering, medicine) where references appear in a dedicated reference section and are referenced in the running text in some form, for example, author-year mentions. The focus is on the identification of these reference sections, linking author-year mentions in the running text to entries in the reference section and the segmentation of references into individual fields, for example, author, title, year etc.

The next related task, quotation detection, aims to identify reported speech, thought and writing in text (Papay and Padó 2019; Pareti et al. 2013; Scheible et al. 2016). This task is normally constrained to individual texts and a focus on speech. For our task on the other hand, we are interested in the detection of quotations as a type of scholarly citation.

Quotation attribution is the task of identifying the source of a quotation (Almeida et al. 2014; Elson and McKeown 2010). Existing approaches are often focused on speaker attribution in fiction or news paper articles. For the task at hand, our goal is different. We want to distinguish between quotations from a given primary literary work and other sources and identify a specific occurrence in the case of multiple occurrences. We aim to combine aspects of these three tasks into the new task of identifying quotations in one text and linking those quotations to their source in another text by using page references.

Arnold and Jäschke (2021) presented Quid: a tool for the identification of text reuse with a focus on quotations with a length of at least five words between literary and scholarly works. Five words is not a hard limit but they determined that shorter quotations generate too many ambiguous matches without more advanced methods. Quid outperformed other approaches which led us to the decision to use it in this work. We will also use Quid for the extraction of candidates for quotations shorter than five words which we then filter further.

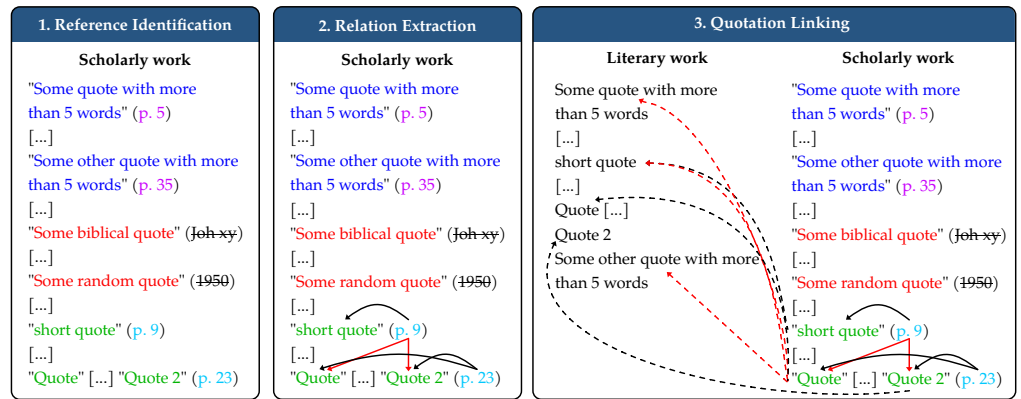


Figure 2: Visualization of quotation identification and linking in three steps.

3. Methods

In this section, we first define the task and then present two approaches to solve it. The first approach is a specialized pipeline and the second approach is a more general approach based on a neural network language model.

3.1 Task

Our overall goal is to identify short quotations in the scholarly work and link the quotations to their source text in the literary work. For this task, we make the following assumptions. Firstly, we work with a corpus of scholarly works for which we know that their main focus is on the primary literary work which we are interested in. Secondly, we assume that all quotations appear in quotation marks and that the texts do not contain errors, for instance, due to OCR. Handling texts with such issues is out of scope of this work and there are other efforts solving this task (Brunner et al. 2020).

We focus on scholarly works with references in parentheses in the running text. This decision was made based on the number of scholarly works in our corpus with references in the running text (subsection 4.1) and due to the high variance in structure of references in footnotes.

3.2 ProQuo

Figure 2 shows the building blocks of our first approach which we named ProQuo. This approach is divided into three steps: *Reference Identification*, *Relation Extraction*, and *Quotation Linking*. In the first step, we use long quotations (dark blue), extracted using Quid, and their references (pink) as anchors. We use these known references as examples to identify other references to the literary work (light blue) and distinguish them from other text in parentheses (strikethrough). In the second step, we then link the identified references to their corresponding short quotations (green). In the final step, the identified short quotations are linked to their source in the literary work (black dashed arrows).

No.	Example	Citation Target
1	(12)	Literary work
2	(12, 12-14)	
3	(S. 12)	
4	(HKA V,1. S.12)	
5	(I, S.12)	
6	(Jb, 12)	
7	(SW9 II, 12)	
8	(1987)	Other
9	(Johannes 8, 11)	
10	(other text)	

Table 1: Examples for references.

3.2.1 Step 1: Reference Identification

The goal of this step is to distinguish between true references to the literary work (Figure 2, light blue) and other text in parentheses. References are written in a number of ways, as Table 1 shows.² Often references only contain a page number (Ex. 1, 3) but they can also contain line numbers (Ex. 2) or information on the cited edition (Ex. 4). In this work, we are only interested in page numbers and ignore the other information. To extract the page number from a reference string, we perform the following searches until we get a match:

- A number which immediately follows the string “S. ”;
- A number which is not preceded by a letter.

A scholarly work can use any of the variants from Table 1 to reference the literary work and at the same time use some other variant to point to other (literary) works or even use a similar looking variant to reference the same source in case of citations from collected works. At the same time, we need to distinguish true references from other text which appears in parentheses. This includes dates (Ex. 8), other citations, for example, Bible citations (Ex. 9), and text in general.

To overcome these challenges, we use the following approach. We first identify the best example for a reference to the literary work in the scholarly work. We use quotations longer than five words (Figure 2, dark blue) to extract up to n_{ref} examples of the type of reference (Figure 2, pink) for a specific scholarly work. The examples are extracted starting with the longest quotation with a maximum distance of d_{ref} characters between reference and quotation and a maximum reference length of l_{ref} . If less than three examples could be found, we use the one from the longest quotation. Otherwise, all examples are clustered with spectral clustering into two clusters. We use the probability that two references are similar (the model to determine similarity is described below) as the similarity in the affinity matrix for the clustering. From the bigger cluster, we then select the reference example which belongs to the longest quotation. This clustering procedure is necessary to reduce the probability of selecting an incorrect reference example, which could happen in cases where Quid made a mistake or when the long

² Examples taken from real texts are shown in the original language. Translations: “S.” → “p.”, “Johannes” → “John”. Other translations are given in the text in brackets.

quotation is not followed by a reference but by some other text in parentheses.

To classify whether two references are similar, we trained a twin network (Bromley et al. 1993) for binary classification. The network is made up of two sub-networks, each a character-level BiLSTM (Hochreiter and Schmidhuber 1997) on top of an embedding layer. The outputs of the sub-networks are compared using Manhattan distance. Two references are classified as similar if the probability given by the model is over a threshold t_{ref} . Using this model, all text in parentheses is compared against the selected example to distinguish between true references and other text occurring in parentheses.

3.2.2 Step 2: Relation Extraction

The goal of this step is to identify relations between quotations, that is, text in quotation marks, and the references identified in the previous step. First, we extract all quotations and create all possible combinations of quotations and references where the quotation and reference are within a distance of d_{rel} tokens. We determine tokens by white space tokenization. We surround the quotation which we are interested in with a start and end tag, replace the reference text with a special tag and also replace all other references with another special tag. Then, we use a machine learning model to classify each pair as belonging together (Figure 2, solid black arrows) or not (for example, *Quote/Quote 2* and *p. 9* (solid red arrows)). We classify a quotation and reference as belonging together if the probability given by the model is over the threshold t_{rel} . For quotations with multiple reference candidates, we take the relation with the highest probability.

For the classification we compare two machine learning models: a token-level BiLSTM with a classification layer with sigmoid activation and a fine-tuned German uncased BERT model³ (Devlin et al. 2019) with a linear layer on top of the pooled output.

3.2.3 Step 3: Quotation Linking

The goal of this step is to link quotations from the scholarly work to their source in the literary work (Figure 2, dashed black arrows) and exclude other possible candidates (dashed red arrows). The main idea is to use long quotations with known links and references as anchors. We then link short quotations relative to these known positions.

Scholarly works cite different editions of the literary work. Since automatic identification of the cited edition is out of the scope of this work, we decided to map all citations to one edition. To achieve this, we estimate a *virtual page* size by using the references from the long quotations:

$$page_size = \frac{last_quote_end - first_quote_start}{last_page - first_page} \quad (1)$$

Here *page_size* is the estimated page length (the number of characters) of the literary work, *first_quote_start* and *last_quote_end* are character positions of the first and last quotation in the scholarly work, respectively, and *first_page* and *last_page* are the corresponding page numbers in the literary work, respectively.

3. See: <https://huggingface.co/dbmdz/bert-base-german-uncased>.

Using this virtual page size we can approximate the character position of short quotations in the literary work. It should be noted that short quotations can appear without a reference. We distinguish between short quotations with and without a reference and the approach differs:

$$page_diff = quote_page - first_page \quad (2)$$

$$quote_pos = first_quote_start + (page_diff \times page_size) \quad (3)$$

For *quotations with a reference*, we approximate the character position of the quotation in the literary work by using Equations 2 and 3, where *quote_page* is the page number of the quotation we want to link and *page_diff* is the distance in number of pages between the quotation and the first known page number.

For *quotations without a reference*, we first try to find the closest quotation in the scholarly work from the already linked quotations within a certain distance d_{link} . We use the midpoint of that quotation as the approximate position.

If an approximate position could be determined, we use this position to define a search range r_{link} . For single word quotations, we then first perform exact string matching and if that does not lead to any matches, we perform fuzzy matching. In case of multiple matches, we take the match closest to the approximated quote position.

For longer quotations, we first try to find an exact match in the determined range. If that leads to exactly one match, that match is used. If there are no matches, the whole text is searched. If that does not lead to a single exact match, we use the matches from Quid as candidates. If there is a single candidate in the given range with an overlap of at least o_{link} %, that candidate is used. If there are no matches, the whole text is searched for a single unambiguous result.

If no approximate position could be determined, the whole text is searched for a single exact match and if there are no matches, we perform fuzzy matching and only use a single unambiguous result.

In our corpus, 11 scholarly works cite an edition of *Michael Kohlhaas* in parallel print. These texts were manually identified and this information is passed to the algorithm to adjust the calculations to only count every other page.⁴

3.3 ProQuoLM

Having seen and appreciated the complexity of the aforementioned bespoke approach, we want to analyze how state-of-the-art neural language models can solve the task when it is formulated in a very simple way such that they can be fine-tuned and applied.

For the second approach, we first extract all text in quotation marks and for each quotation we determine all candidates in the literary work. For determining the candidates we use the same (fuzzy) matching approach as in [subsubsection 3.2.3](#). We then fine-tune the same German uncased BERT model as before for binary classification between a quotation and a candidate, both with a context window. Both text fragments, that is, quotation with context and candidate with context, have a maximum length of l_m tokens

4. This is just a very rough approximation. The topic of parallel editions is much more complex and beyond the scope of this work.

Literary work	Die Judenbuche	Michael Kohlhaas
All quotations (primary work)	1 736	1 788
Quotations with a reference	1 467	1 547
Short quotations	817	862
Short quotations with a reference	672	736
Quotations in footnotes	94	80

Table 2: Statistics for *Die Judenbuche* and *Michael Kohlhaas*.

each. We also surround the quotation which we are interested in with a start and end tag in both fragments. From all candidates, we select the one with the highest probability over a threshold t_{lm} .

4. Experiments

In this section, we first give an overview of the dataset and our annotations. We then present the experiments to evaluate both approaches on texts with references in the running text. Finally, we evaluate ProQuoLM on texts with all reference information removed.

4.1 Dataset and Annotation

We assess our methods by analyzing two literary texts, *Die Judenbuche* by Annette von Droste-Hülshoff (1979) and *Michael Kohlhaas* by Heinrich von Kleist (1978). For each text, our corpus contains 44 and 49 interpretive scholarly articles, respectively, which were previously annotated in the ArguLIT project (Winko 2017–2020) using TEI/XML (TEI Consortium, eds. 2022).⁵ The annotations include quotations of different types, such as those from the primary literary work, other literary works or scholarly works. The original annotations were limited to clearly marked quotations, that is, with quotation marks. In this evaluation, we only focus on quotations coming from the primary literary work. The 93 scholarly works use references either in parentheses in the running text or in footnotes. For this work, we focus on scholarly works with references in the running text and ignore footnotes in all experiments, including quotations in footnotes. This decision was made mainly due to the varying structure of quotations in footnotes and to keep the focus on quotations with references in the running text. For *Die Judenbuche*, 24 scholarly works and for *Michael Kohlhaas*, 33 scholarly works have references in the running text.

We extended the original annotations of these scholarly works in two annotation tasks. In the *reference annotation task*, three persons annotated reference strings (Table 1) and links between reference strings and quotations. Five of the texts were annotated by all three annotators with F_1 -score inter-annotator agreements between pairs of annotators of 0.88, 0.93, and 0.90. Table 2 shows statistics for the number of (short) quotations from the primary literary work with and without references. We also show the number of quotations in footnotes which only account for around 10 % of short quotations.

In a *linking annotation task*, two persons annotated the origin of quotations from scholarly

5. For the sake of brevity, we will reference *Die Judenbuche* and *Michael Kohlhaas* with J and K, respectively.

texts in the literary text. In this task, not only the literary works with references in the running text were annotated, but for *Die Judenbuche*, all 44 scholarly works were annotated. The additional annotated texts contain 270 short quotations. For consistency, we also ignore footnotes in the additionally annotated texts where references appear in footnotes, effectively resulting in texts without any reference information. This data is used to evaluate the performance of ProQuoLM, which does not rely on explicit reference information, on texts without references. To evaluate inter-annotator agreement, again, the same five texts as before were annotated by both annotators which resulted in an F_1 -score inter-annotator agreement of 0.90.

4.2 References in Running Text

For the experiments in this section, we perform 5-fold cross validation. We calculate precision and recall following Arnold and Jäschke (2021). We optimized the hyperparameters once on the validation data from the first split of our cross validation and use the hyperparameters for all evaluations.

4.2.1 Reference Identification

To evaluate the performance of our model, we compare it against a baseline that classifies texts in parentheses as a reference if there is at least one number contained and the text is not longer than the maximum reference length l_{ref} , which we set to 25 characters. This value was chosen as it is in the 99 percentile of lengths in our corpus

The output dimension of the embedding layer and the BiLSTM hidden state are both 32. A dropout of 0.2 is applied. The batch size was set to 512 and the network was trained for 10 epochs with binary crossentropy loss and Adam optimizer with a learning rate of 0.001. The number of examples n_{ref} is set to 5. This worked well in our tests and leaves some room for incorrect examples. For the maximum distance d_{ref} , we determined 20 characters to work well. The inputs are padded/truncated to the maximum reference length. The classification threshold t_{ref} is set to 0.7.

4.2.2 Relation Extraction

To evaluate the performance of our two models, we compare them against three baselines. The first baseline (*Ref After*) always takes the closest reference after the quotation. The second baseline (*Ref Before*) works the same way but takes the closest references before the quotation and the last baseline (*Ref Closest*) takes the closest reference before or after the quotation.

For the BiLSTM model, the output dimension of the embedding layer is 64, the hidden state is 64, and a dropout of 0.3 is applied. The batch size was set to 128 and the network was trained for 5 epochs with binary crossentropy loss and Adam optimizer with a learning rate of 0.01. We use WordPiece embeddings (Wu et al. 2016) with a 8 000 token vocabulary. The classification threshold t_{rel} is set to 0.4. The BERT model was fine-tuned for 3 epochs with a batch size of 12 and a learning rate of 10^{-5} . The classification threshold t_{rel} is set to 0.5. The maximum distance d_{rel} between a quotation and reference to still be considered is 100 tokens which is in the 93 percentile of distances in our

corpus. We tried to increase the maximum distance but got overall worse results as false positives increased. The input is padded/truncated to a length of 200 tokens.

4.2.3 Quotation Linking

To evaluate the performance of our algorithm, we compare it against a baseline which always links a quotation to the first matching instance.

We determined a search range r_{link} of one page before and after the approximate position to work best. For quotations without a reference, the maximum distance d_{link} is 500 tokens. The minimum candidate overlap o_{link} is 70%.

4.2.4 The Complete Pipeline and Language Model Approach

In this experiment, we perform two evaluations of our two approaches and compare the results against the same baseline as for the quotation linking task. We first perform the same 5-fold cross validation as before and then a second evaluation where we split the scholarly works by the literary work they interpret and train on the texts from one literary work and evaluate on the other. This is relevant as it indicates how well the approaches can generalize and perform on a completely new literary work.

For ProQuoLM, the model was fine-tuned for 3 epochs with a batch size of 4 and a learning rate of 10^{-5} . The classification threshold t_{lm} is set to 0.5 and the maximum length l_{lm} to 200 tokens.

4.3 References in Footnotes

Our second approach ProQuoLM does not rely on explicit reference information for quotations. With this experiment, we investigate whether reference information is needed at all or if our second approach can also handle texts with references in footnotes. We do this by evaluating how well ProQuoLM performs on texts where all reference information is removed including footnotes.

5. Results

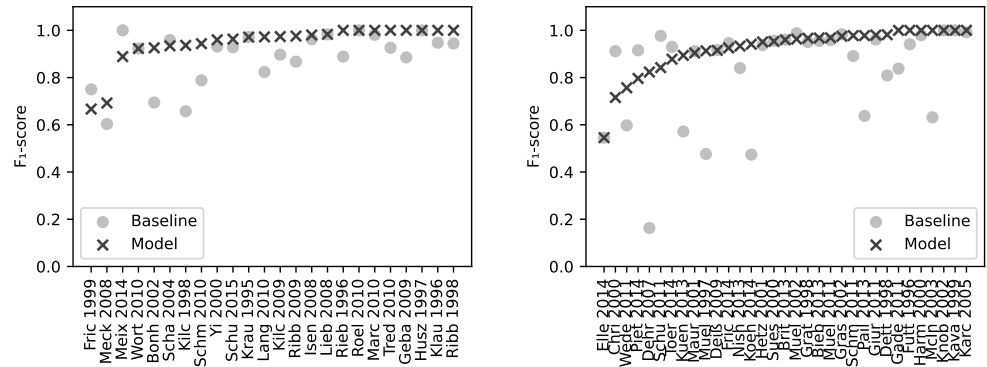
We first present the results for the experiments of the individual steps of ProQuo (subsection 5.1 to subsection 5.3), followed by the results of the complete pipeline ProQuo compared to ProQuoLM (subsection 5.4). Finally, we present how ProQuoLM performs on texts without any reference information (subsection 5.5).

5.1 Reference Extraction

Table 3 shows the results for our baseline and model for reference extraction. Our model outperforms a strong baseline for both literary works. The baseline only misses cases where the reference is not in parentheses or does not contain a number, for example, *ibd.* [ibid.] False positives include dates, Bible quotations, or quotations from other scholarly texts. Our model misses less *ibd.* references but all cases not in parentheses and some other special cases. This includes instances where the reference style differs from all other references, for example, references to a specific verse (*V. 8*) and not a page.

Approach	Die Judenbuche			Michael Kohlhaas		
	Precision	Recall	F ₁	Precision	Recall	F ₁
Baseline	0.86	0.95	0.90	0.80	0.95	0.87
Model	0.95	0.96	0.95	0.97	0.90	0.93

Table 3: Precision, recall, and F₁-score for *Die Judenbuche* and *Michael Kohlhaas* for reference classification.



(a) *Die Judenbuche*

(b) *Michael Kohlhaas*

Figure 3: F₁-score comparison for reference extraction.

Other false negatives include references that consist of two references (*S. 47 und S. 50*) and references which differ from the rest as they are followed by additional information (*Jb, 35, Herv. durch Autor* [author’s emphasis]). False positives include instances where numbers appear in parentheses with the same style as true references but are used to structure the text (e. g., in enumerations) or reference other scholarly works.

Figure 3a and Figure 3b enable a more fine grained analysis.⁶ For *Die Judenbuche*, we can see that our model outperforms or is on par with the baseline for all texts except three. We get similar results for *Michael Kohlhaas*, except that for seven texts the baseline performs better than the model. The results illustrate the importance of our model. Texts for which the baseline struggles often have a high number of quotations with references from sources other than the primary literary work.

5.2 Relation Extraction

Table 4 shows the results of our two models and three baselines. *Ref Closest* is the best performing baseline with an F₁-score of 0.65 (*Die Judenbuche*) and 0.75 (*Michael Kohlhaas*). *Ref Closest* has the highest recall but lacks precision. This is to be expected as the baseline does not distinguish between quotations from the primary literary work and quotations from other sources. The poor performance of *Ref Before* confirms that references typically follow a citation.

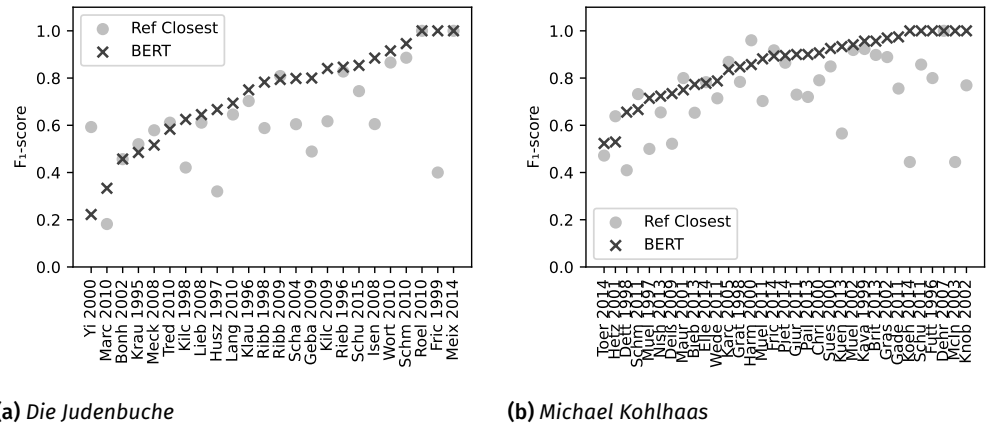
The LSTM-based model outperforms all three baselines. The BERT model performs best overall but worse for *Die Judenbuche* than for *Michael Kohlhaas*.

For *Die Judenbuche*, there are 213 false negatives; 98 of those are the result of long

6. The horizontal axes are labeled with the first (up to four) letters of the first author’s name followed by the year of publication. The labels can be used to identify the texts on: <https://hu.berlin/quidex>.

Approach	Die Judenbuche			Michael Kohlhaas		
	Precision	Recall	F ₁	Precision	Recall	F ₁
Ref After	0.59	0.63	0.61	0.72	0.78	0.75
Ref Before	0.25	0.21	0.23	0.14	0.12	0.13
Ref Closest	0.57	0.76	0.65	0.66	0.86	0.75
LSTM	0.83	0.59	0.69	0.85	0.69	0.76
BERT	0.83	0.68	0.74	0.93	0.81	0.86

Table 4: Precision, recall, and F₁-score for *Die Judenbuche* and *Michael Kohlhaas* for relation extraction.



(a) *Die Judenbuche*

(b) *Michael Kohlhaas*

Figure 4: F₁-score comparison for relation extraction.

distances, that is, the distance between quotation and references is larger than 100 tokens. Another 67 are instances where the reference appears before the quotation. We get a similar result for *Michael Kohlhaas* with 178 false negatives, 69 long distance and 76 reference before quotation. The instances where the reference appears before the quotation are problematic due to the fact that a reference before a quotation is a lot less likely and our training data is limited in that regard.

Figure 4a and Figure 4b show a comparison of the best baseline and the BERT model. These results illustrate the importance of the model for the difficult texts where the difference in performance between the baseline and model is largest. But they also show that the model struggles with some texts. In the case of *Yi 2000*, for example, all false positives are instances where the reference appears before the quotation.

5.3 Quotation Linking

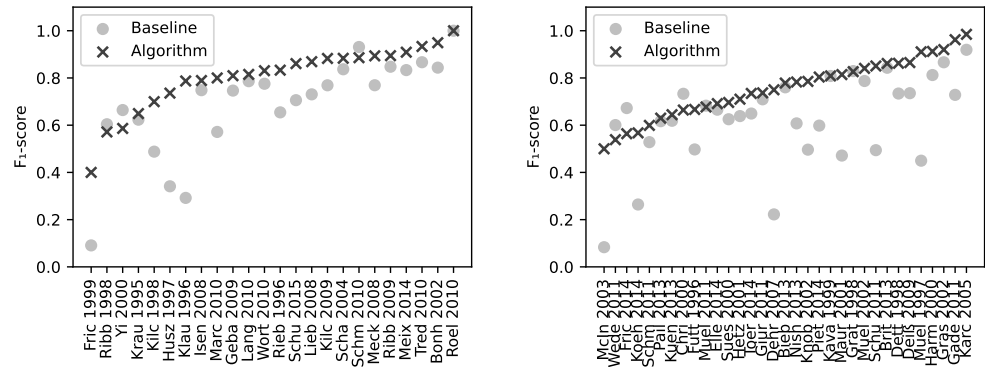
Table 5 shows the results for the quotation linking step. We compare our algorithm against one baseline (see also Figure 5a and Figure 5b).

The algorithm outperforms the baseline for both literary works and achieves a high precision. The baseline struggles with texts with a low percentage of quotations from the primary literary work which still appear in the literary work.

Our algorithm generates 158 false negatives for *Die Judenbuche*. 102 of those are single word quotations and 111 have a reference in our annotations. For *Michael Kohlhaas*, we get 271 false negatives of which 180 are single word quotations and 215 have a

Approach	Die Judenbuche			Michael Kohlhaas		
	Precision	Recall	F ₁	Precision	Recall	F ₁
Baseline	0.65	0.77	0.70	0.59	0.74	0.66
Algorithm	0.85	0.77	0.81	0.86	0.69	0.76

Table 5: Precision, recall, and F₁-score for *Die Judenbuche* and *Michael Kohlhaas* for quotation linking.



(a) *Die Judenbuche*

(b) *Michael Kohlhaas*

Figure 5: F₁-score comparison for quotation linking.

reference. These results indicate that for further improvements better handling of single word quotations is necessary. The results for *Die Judenbuche* would also indicate that an improvement in the relation extraction step should improve the overall results of the pipeline. At first glance, the overall worse results for *Michael Kohlhaas* in combination with the better results in the relation extraction step do not support this theory. But *Michael Kohlhaas* is roughly twice as long as *Die Judenbuche*, which makes the linking step considerably harder and which could counteract the better relation extraction performance.

5.4 The Complete Pipeline and Language Model Approach

The results in Table 6 demonstrate that both approaches – ProQuo and ProQuoLM – perform on the same level. Compared to the baseline, the pipeline is a big improvement in precision, but recall is lower than the baseline for both literary works. Overall, ProQuoLM works best, with improvements in recall over ProQuo. ProQuoLM produces 169 false negatives for *Die Judenbuche*, 104 are single word quotations and 133 have a reference in our annotations. Similarly for *Michael Kohlhaas*, the results contain 267 false negatives, 177 are single word quotations and 222 have a reference.

The second evaluation shows the performance of ProQuo and ProQuoLM for training and evaluation split by literary work. This is relevant as it indicates what the performance will be on a completely new literary work. We can see that the difference in performance is larger when the scholarly works from *Die Judenbuche* are used as training data. This is not surprising as there are less scholarly works for *Die Judenbuche* and therefore less training data in that case.

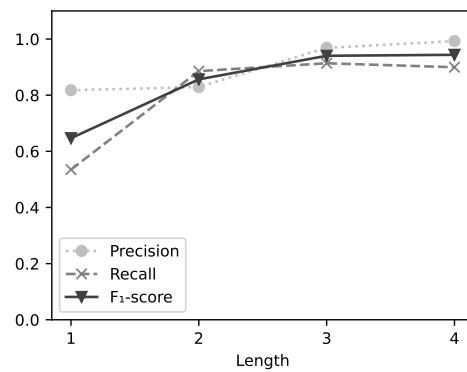
In Figure 6a–Figure 6d, we report results broken down by quotation length in words.

Approach	Die Judenbuche			Michael Kohlhaas		
	Precision	Recall	F ₁	Precision	Recall	F ₁
Baseline	0.65	0.77	0.70 [0.60, 0.78]	0.59	0.74	0.66 [0.56, 0.69]
ProQuo	0.87	0.72	0.79 [0.73, 0.82]	0.87	0.66	0.75 [0.69, 0.78]
ProQuoLM	0.88	0.74	0.80 [0.74, 0.86]	0.86	0.69	0.77 [0.70, 0.81]

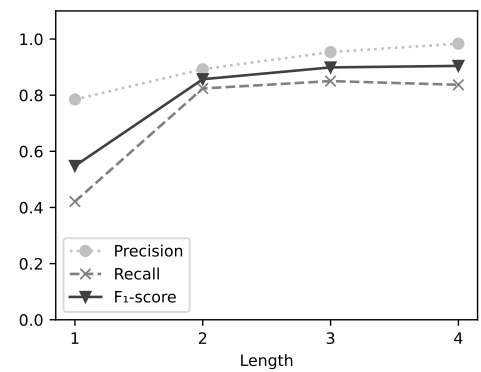
Split by literary work						
ProQuo	0.87	0.71	0.78 [0.72, 0.82]	0.85	0.63	0.72 [0.66, 0.76]
ProQuoLM	0.82	0.73	0.77 [0.70, 0.82]	0.75	0.70	0.72 [0.65, 0.77]

Table 6: Precision, recall, and F₁-score for *Die Judenbuche* and *Michael Kohlhaas* for the full pipeline. For each F₁-score, the upper and lower bound of the 95 % confidence interval is reported.

ProQuo

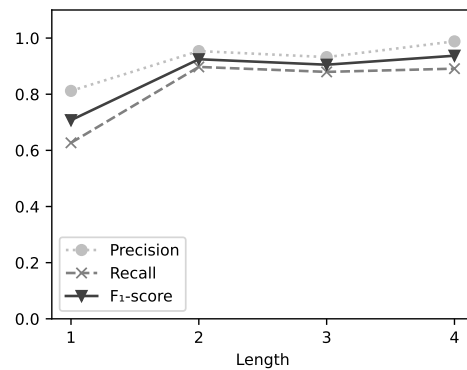


(a) *Die Judenbuche*

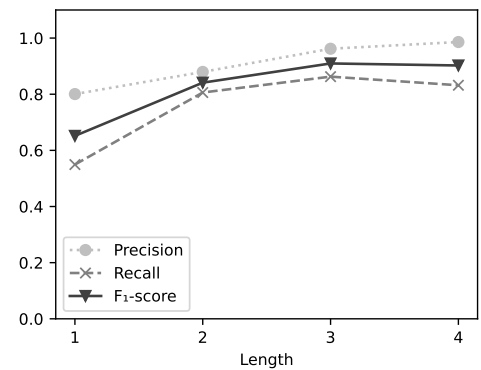


(b) *Michael Kohlhaas*

ProQuoLM



(c) *Die Judenbuche*



(d) *Michael Kohlhaas*

Figure 6: Precision, recall, and F₁-score for *Die Judenbuche* and *Michael Kohlhaas* by length (in words) of quotations.

For both tools, ProQuo (top) and ProQuoLM (bottom), and both literary works, *Die Judenbuche* (left) and *Michael Kohlhaas* (right), single word quotations are the most difficult to identify and link. Similarly, both tools achieve better results for quotations of length three and four. Interestingly, for *Die Judenbuche* and two word quotations, there is a substantial difference in precision between the two approaches. We found that this is due to the term *Die Judenbuche* which ProQuo incorrectly identifies as a quotation in a number of cases. If we exclude these false positives, the precision rises from 0.83 to 0.93.

Approach	Precision	Recall	F ₁
Baseline	0.52	0.86	0.65
ProQuoLM	0.80	0.83	0.81

Table 7: Precision, recall, and F₁-score for *Die Judenbuche* for texts with references in footnotes.

The results show that the performance of both tools is on the same level, but from a usability perspective, ProQuoLM is superior to ProQuo. The approach is less complex, the creation of training data is a lot less time consuming and there is no need for specific handling of parallel print editions.

5.5 References in Footnotes

In this final experiment, we evaluate the performance of ProQuoLM trained solely on scholarly texts from *Michael Kohlhaas* and tested on scholarly texts from *Die Judenbuche* with references in footnotes. But, as before, we exclude footnotes, effectively resulting in scholarly works without any reference information. We compare ProQuoLM against the same baseline as before.

[Table 7](#) shows that the performance is similar to the other results. This means that even without reference information, ProQuoLM performs on the same level as ProQuo which further highlights its advantages, as it is more versatile. It also leads us to the conclusion that ProQuoLM currently cannot make use of the information contained in references. Considering that there is no information available to the model from where in the literary work a candidate is taken, this is not surprising. Another reason could be that BERT is struggling with capturing numeracy (Wallace et al. 2019).

6. Discussion

We presented two approaches for the identification and linking of short quotations between scholarly works and literary works. ProQuo is a pipeline consisting of three steps. We evaluated each step individually as well as the complete pipeline. ProQuo outperforms a strong baseline, which lacks precision, especially in cases with quotations from different sources. Our results illustrate that the simple approach of just performing text matching is not sufficient for the task at hand.

The second approach, ProQuoLM, performs on the same level as the pipeline but is superior from a usability perspective as it is less complex, more versatile and the creation of training data is less time consuming. We therefore consider ProQuoLM to be a better starting point for future improvements. However, it should be noted that, depending on the overall goal, ProQuo has the advantage that the idea behind the overall approach and the individual steps can be explained which makes it easier to identify specific issues. The following observations might not have been made without the pipeline and the possibility to investigate individual steps. The development of two approaches is more time and resource consuming but can be beneficial.

From our experiments, we can observe a number of things. Firstly, the distance between a quotation and corresponding reference information can be quite large but our context

window is limited due to limitations of current language models. Secondly, the quotation linking step struggles with single word quotations even if they come with a reference. Lastly, ProQuoLM performs on the same level with and without reference information. Based on these observations alone it is not possible to determine the exact source of the remaining issues without further experiments. As a first step, we propose to test ProQuoLM with positional information from where a candidate is taken in the literary work to see if ProQuoLM can make use of reference information at all in the current version. Additionally, it might also be the case that more training would already improve this approach. Also, explicit usage of reference information from the first step of the pipeline in combination with ProQuoLM could be promising but, again, is limited by the fact that reference information can be scattered throughout the text.

Other areas for improvement include the resolution of references which point to other references, for example *ibid.*, and references with multiple page numbers, page ranges or line numbers which are currently not properly handled. We also do not handle quotations with multiple occurrences in the literary work. In the current approach, quotations are never linked to more than one occurrence.

For the presented approaches, we assume a corpus of scholarly works for which we know that the main source of quotations is a certain literary work. Arnold and Jäschke (2022) have found that existing approaches for automatic extraction of bibliographic information do not work for scholarly works in literary studies. This led us to conclude that advances in the extraction of literature references are needed before we can make use of bibliographic information to automatically match scholarly works with the main literary work in focus. Advances in this area would also allow for proper handling of citations from different editions of the literary work.

Another assumption we made for this work is that all quotations appear in quotation marks and that the texts do not contain errors, for instance, due to OCR or mistakes made by the authors. We did not analyze how such errors influence the results as it is beyond the scope of this work. Based on our findings, it seems likely that these errors would have a bigger impact on ProQuo compared to ProQuoLM considering that the former relies more on the availability of specific information. But a deeper analysis is needed to come up with quantifiable results.

7. Data Availability

The annotated scholarly works can currently not be made available due to copyright restrictions. All data that can be made available can be found here: <https://hu.berlin/proquo-resources> (DOI: <https://doi.org/10.5281/zenodo.8232596>).

8. Software Availability

Software can be found here: <https://hu.berlin/proquo> (DOI: <https://doi.org/10.5281/zenodo.8221381>)

9. Acknowledgements

Parts of this research were funded by the German Research Foundation (DFG) priority programme (SPP) 2207 *Computational Literary Studies* project *What matters? Key passages in literary works* (grant no. 424207720). We would like to thank the project's student assistants Gregor Sanzenbacher and Nathalie Burkowski and our colleague Benjamin Fiechter for their annotation work as well as Steffen Martus for giving feedback on the manuscript.

10. Author Contributions

Frederik Arnold: Software, Experiments, Conceptualization, Writing – original draft

Robert Jäschke: Conceptualization, Writing – original draft

References

- Almeida, Mariana S. C., Miguel B. Almeida, and André F. T. Martins (2014). "A Joint Model for Quotation Attribution and Coreference Resolution". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 39–48. [10.3115/v1/E14-1005](https://doi.org/10.3115/v1/E14-1005).
- Arnold, Frederik and Benjamin Fiechter (2022). "Lesen, was wirklich wichtig ist - Die Identifikation von Schlüsselstellen durch ein neues Instrument zur Zitatanalyse". In: *DHd 2022 Kulturen des digitalen Gedächtnisses. 8. Tagung des Verbands Digital Humanities im deutschsprachigen Raum*. DHd-Verband. [10.5281/zenodo.6327917](https://doi.org/10.5281/zenodo.6327917).
- Arnold, Frederik and Robert Jäschke (2021). "Lotte and Annette: A Framework for Finding and Exploring Key Passages in Literary Works". In: *Proceedings of the Workshop on Natural Language Processing for Digital Humanities*. NLP Association of India (NLPAI), 55–63. <https://aclanthology.org/2021.nlp4dh-1.7> (visited on 11/02/2023).
- (2022). "A Game with Complex Rules: Literature References in Literary Studies". In: *Proceedings of the Workshop on Understanding Literature references in academic full TExt*. CEUR Workshop Proceedings, 7–15. <https://ceur-ws.org/Vol-3220/paper1.pdf> (visited on 11/02/2023).
- Bloomfield, Lou (2016). *Copyfind*. <https://plagiarism.bloomfieldmedia.com/software/copyfind/> (visited on 11/02/2023).
- Bromley, Jane, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah (1993). "Signature Verification Using a Siamese Time Delay Neural Network". In: *Advances in Neural Information Processing Systems*. Vol. 6. <https://dl.acm.org/doi/10.5555/2987189.2987282>.
- Brunner, Annelen, Ngoc Duyen Tanja Tu, Lukas Weimer, and Fotis Jannidis (2020). "To BERT or not to BERT-Comparing Contextual Embeddings in a Deep Learning Architecture for the Automatic Recognition of four Types of Speech, Thought and Writing Representation." In: *SwissText/KONVENS*. <https://ceur-ws.org/Vol-2624/paper5.pdf> (visited on 11/02/2023).
- Da, Nan Z. (2019). "The Computational Case against Computational Literary Studies". In: *Critical Inquiry* 45 (3), 601–639. [10.1086/702594](https://doi.org/10.1086/702594).

- Descher, Stefan and Thomas Petraschka (2018). “Die Explizierung des Impliziten”. In: *Scientia Poetica* 22 (1), 180–208. [10.1515/scipo-2018-007](https://doi.org/10.1515/scipo-2018-007).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186. [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423).
- Droste-Hülshoff, Annette von (1979). *Die Judenbuche*. Insel Verlag. <https://www.projekt-gutenberg.org/droste/judenbch/index.html> (visited on 11/02/2023).
- Elson, David K. and Kathleen R. McKeown (2010). “Automatic Attribution of Quoted Speech in Literary Narrative”. In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. AAAI’10. Atlanta, Georgia: AAAI Press, 1013–1019.
- GROBID (2008–2022). <https://github.com/kermitt2/grobid>. swb: 1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long Short-Term Memory”. In: *Neural Computation* 9 (8), 1735–1780. [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- Hohl Trillini, Regula and Sixta Quassdorf (2010). “A ‘key to all quotations’? A corpus-based parameter model of intertextuality”. In: *Literary and Linguistic Computing* 25 (3), 269–286. [10.1093/l1c/fqq003](https://doi.org/10.1093/l1c/fqq003).
- Kleist, Heinrich von (1978). “Michael Kohlhaas”. In: *Werke und Briefe in vier Bänden*. Ed. by Michael Holzinger. CreateSpace Independent Publishing Platform, 7–113. <http://www.zeno.org/nid/2000516902X> (visited on 11/02/2023).
- Molz, Johannes (2020). *A Close and Distant Reading of Shakespearean Intertextuality: Towards a Mixed Method Approach for Literary Studies*. Open Publishing in the Humanities. Universitätsbibliothek Ludwig-Maximilians-Universität. [10.5282/oph.4](https://doi.org/10.5282/oph.4).
- Papay, Sean and Sebastian Padó (2019). “Quotation Detection and Classification with a Corpus-Agnostic Model”. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. INCOMA Ltd., 888–894. [10.26615/978-954-452-056-4_103](https://doi.org/10.26615/978-954-452-056-4_103).
- Pareti, Silvia, 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*. Association for Computational Linguistics, 989–999. <https://aclanthology.org/D13-1101> (visited on 11/02/2023).
- Prasad, Animesh, Manpreet Kaur, and Min-Yen Kan (2018). “Neural ParsCit: a deep learning-based reference string parser”. In: *International Journal on Digital Libraries* 19 (4), 323–337. [10.1007/s00799-018-0242-1](https://doi.org/10.1007/s00799-018-0242-1).
- Reeve, Jonathan (2020). *JonathanReeve/text-matcher: First Zenodo release*. Zenodo. version 0.1.6. [10.5281/zenodo.3937738](https://doi.org/10.5281/zenodo.3937738).
- Schaum, Konrad (2004). *Ironie und Ethik in Annette von Droste-Hülshoffs Judenbuche*. Beiträge zur neueren Literaturgeschichte; [Folge 3], Bd. 204. Winter. Chap. Die Judenbuche als Sittengemälde, 99–194.
- Scheible, Christian, Roman Klinger, and Sebastian Padó (2016). “Model Architectures for Quotation Detection”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1736–1745. [10.18653/v1/P16-1164](https://doi.org/10.18653/v1/P16-1164).

- Smith, David A., Ryan Cordell, Elizabeth Maddock Dillon, Nick Stramp, and John Wilkerson (2014). "Detecting and Modeling Local Text Reuse". In: *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL '14. London, United Kingdom: IEEE Press, 183–192.
- TEI Consortium, eds. (2022). *TEI P5: Guidelines for Electronic Text Encoding and Interchange, Version 4.4.0*. <https://www.tei-c.org/Guidelines/P5/> (visited on 04/29/2022).
- Wallace, Eric, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner (2019). "Do NLP Models Know Numbers? Probing Numeracy in Embeddings". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 5307–5315. [10.18653/v1/D19-1534](https://doi.org/10.18653/v1/D19-1534).
- Winko, Simone (2017–2020). *The making of plausibility in interpretive texts. Analyses of argumentative practices in literary studies*. DFG-funded research project (grant no. 372804438). <https://gepris.dfg.de/gepris/projekt/372804438?language=en> (visited on 11/02/2023).
- Winko, Simone and Fotis Jannidis (2015). "Wissen und Inferenz – Zum Verstehen und Interpretieren literarischer Texte am Beispiel von Hans Magnus Enzensbergers Gedicht Frühschriften". In: *Literatur interpretieren: Interdisziplinäre Beiträge zur Theorie und Praxis*. Ed. by Jan Borkowski, Stefan Descher, Felicitas Ferder, and Philipp David Heine. Brill | mentis, 221–250. [10.30965/9783957438973](https://doi.org/10.30965/9783957438973).
- Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. [10.48550/ARXIV.1609.08144](https://arxiv.org/abs/1609.08144).