



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

MASTER THESIS

Evgeniya Ustinova

Automatic detection and attribution of quotes

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: RNDr. Jiří Hana, Ph.D.

Study programme: Computer Science

Study branch: Language Technologies and
Computational Linguistics

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date
Author's signature

My master's degree journey and thesis journey were quite long, but it would have been much more difficult without many people who supported me on my way and to whom I would like to express my gratitude:

To my supervisors Jirka Hana and Claudia Borg. In the moments when I felt overwhelmed with the amount of work, their guidance helped me to stay on track.

To Silvia Pareti for providing the PARC dataset and being kind to answer my small questions.

To Markéta Lopatková, Claudia Borg, Anna Felsing, Bobbye Pernice and other people who ran and run LCT master's program. Your work is very valuable.

To LCT fellows, who were always ready to listen to me and relieve my anxiety, especially to Jacob, Katya, Anya and Cyrill, спасибо!

I appreciate being a scholarship holder from the Erasmus+ Programme of the EU. I do not exaggerate by saying that it was a life-changing opportunity.

My deepest gratitude goes to my husband whose constant support, helped me tremendously to accomplish this work.

Title: Automatic detection and attribution of quotes

Author: Evgeniya Ustinova

Institute: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Jiří Hana, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Quotations extraction and attribution are important practical tasks for the media, but most of the presented solutions are monolingual. In this work, I present a complex machine learning-based system for extraction and attribution of direct and indirect quotations, which is trained on English and tested on Czech and Russian data. Czech and Russian test datasets were manually annotated as part of this study. This system is compared against a rule-based baseline model. Baseline model demonstrates better precision in extraction of quotation elements, but low recall. The machine learning-based model is better overall in extracting separate elements of quotations and full quotations as well.

Keywords: quotation extraction, quotation attribution, CRFs, article, annotation

Contents

Introduction	3
1 Background	5
1.1 About Quotations	5
1.2 Quotations Extraction and Attribution Tasks	9
1.3 Previous Research	10
1.3.1 Studies on English News Texts	10
1.3.2 Studies on Literary Texts and Other Languages	11
1.4 Datasets	12
1.5 Metrics	13
1.6 Description of CRFs	15
2 Experiments	17
2.1 Datasets	17
2.1.1 English Data	17
2.1.2 Czech Data	19
2.1.3 Russian Data	20
2.2 Standardization of Datasets	22
2.3 Evaluation Methods	24
2.4 Implementation Details	25
2.5 Baseline Model	26
2.6 Machine Learning-Based System	28
2.7 Features Analysis	32
3 Results & Discussion	35
3.1 Baseline Model	35
3.1.1 English	35
3.1.2 Czech	36
3.1.3 Russian	36
3.2 Machine Learning-Based System	37
3.3 Evaluation on the SiR Dataset	40
3.4 Trained Models Analysis	41
3.4.1 Content Classifier Analysis	41
3.4.2 Source Classifier Analysis	43
3.5 Discussion	44
Conclusion	47
Bibliography	49
List of Figures	53
List of Tables	55
List of Abbreviations	57

A	Lists of reporting verbs	59
A.1	List of English reporting verbs	59
A.2	List of Czech reporting verbs	60
A.3	List of Russian reporting verbs	61
B	Electronic attachments	63
B.1	Requirements	63
B.2	Usage	63
	B.2.1 Quotes Extraction Tool	63
	B.2.2 Quotes Extraction Experiments	64
B.3	Developer documentation	64

Introduction

Quotations are a critical component of media texts, and their proper use is essential in ensuring journalistic integrity and accuracy. In the media, quotes are used to provide insight, context, and support to news stories and other forms of media content. They are essential in providing an accurate representation of the views, opinions, and attitudes of those who are quoted. Furthermore, quotes are often used to give a voice to those who may not otherwise have a platform, such as marginalized communities or individuals. Quotations are also essential in providing transparency and accountability in the media, as they allow readers to evaluate the sources and credibility of the information presented.

However, the use of quotations in the media is not without its challenges. Mis-attributed or out-of-context quotes can mislead readers, undermine journalistic credibility, and have serious consequences for those who are quoted. Therefore, it is essential to ensure that quotes are accurately attributed, contextualized, and used ethically. Automatic detection and attribution of quotes can help improve the accuracy and transparency of media texts while reducing the risk of misrepresentation and misinformation.

Large media companies are interested in automated solutions for quotation detection and attribution problems. The BBC Research&Development department published their research on this question [Newell et al., 2018a], and The Guardian journalists are also working in this direction. The idea for my work came from the article in The Guardian about the project where journalists developed a solution to extract quotations [Guardian, 2021].

The first-ever system for automatic source-quotation extraction was presented by the Joint Research Centre of European Commission [Pouliquen et al., 2007]. It was a rule-based system for direct quotations, but it worked for 11 languages. Since then, most of the work in this area has been done on English data, and most of the relevant datasets are also in English. I decided to build a model that would be trained on English data, but also work on Czech and Russian texts.

In this work, I aim to develop a language-agnostic system that automatically detects direct and indirect quotes from news articles. For illustration, the system should be able to extract *that climate change is a huge concern for the future* and connect it to *The scientist* using a word *said* from the sentence *The scientist said that climate change is a huge concern for the future*. For this, I analyze different types of quotations and create Czech and Russian datasets of attributed quotations to test the model.

I develop a baseline model and a machine learning-based system that are trained on English and then tested on English, Czech, and Russian. The machine learning-based system is founded upon the work of Newell et al. [2018a]. It uses Conditional Random Fields (CRFs) model for its main components as it was used in previous works [Pareti et al., 2013].

The work is structured as follows: in Chapter 1, I overview different types of quotations and give definitions of quotation extraction and attribution task. In the same chapter, I provide an overview of the previous research, in particular, which datasets exist for these tasks and which metrics are used for evaluation. I finish the chapter with a theoretical explanation of how the Conditional Random

Fields model works.

Chapter 2 contains all information about conducted experiments. It starts with a description of the used datasets and the evaluation scheme. Next, I describe the setup of experiments and write about the caveats of running experiments on the Czech and Russian datasets.

Chapter 3 contains all quantitative results of experiments and a discussion of the possible interpretations of the models' performance. In conclusion, I provide a summary of the work and key findings and outline potential future research directions in this field.

The source code for all experiments and quotes extraction tool is in the electronic attachment to this thesis and online.¹

¹<https://github.com/pixelmagenta/adaq>

1. Background

In this chapter, I explain what a quotation is and give an overview of different types of quotations. Then I define a quotation extraction and attribution task and tell about previous work on this topic. Examples in this chapter are provided from the corpora which I use in my research (PARC, UD PDT, UD SynTagRus) and from media. In separate sections, I present the existing datasets for this task and the metrics that are used. In addition, I provide a theoretical description of a CRFs model.

1.1 About Quotations

Quotations can be broadly categorized into two types: direct and indirect. Direct quotations reflect the agent’s speech act (see Example (1)), while indirect quotations report speech acts from the reporter’s perspective (see Example 2). Strictly speaking, the latter is no longer a quotation, but the term is still used this way. In an indirect quotation, the reporter can give more of their analysis on the reported utterance or event.

- (1) “Climate change is a huge concern for the future,” said the scientist.
- (2) The scientist said that climate change is a huge concern for the future.

In my study, I rely on three elements of the quotation. The following terms were first introduced by Pareti [2012b] and became common terminology for the quotations extraction task.

- **Source** is an entity or a person who is the author of the statement. In Example (1), it is **the scientist**. I use bold to highlight it.
- Cue is a word or phrase signaling the connection between the source and the utterance. Usually, it is a verb or verb phrase, but it can also be other parts of speech (e.g., *his statement was*). In the Example (1), it is said. I underline cues in the examples.
- *Content* is the essential element: the message itself. Tokens inside quotation marks in the Example (1) *Climate change is a huge concern for the future*, represent content. I highlight it in italic in further examples.

In most languages, direct quotations are highlighted typographically with quotation marks. They can be of various shapes and orientations. There are double quotation marks in English “...” and Czech (but different orientation) „...“ (see Example (3)) or so-called guillemets «...» in Russian (see Example (4)) and French and reversed guillemets »...« in German (see Example (5)). However, English-style quotation marks are often used in other languages as an alternative style.

Outside quotation marks, punctuation differs as well. If a quotation goes at the end of the sentence, there will be a colon before the opening quotation mark

in Russian.¹ A colon can also be used in English, but it is optional.² In Russian, a comma goes outside quotation marks together with an m-dash. A comma is also located in German after the closing quotation mark [Stang and Steinhauer, 2014]. However, a comma is usually before the closing mark in English and Czech.³

- (3) „Věřím, že nám jako právní poradce pomůže obnovit
I believe that us as legal advisor help restore
 důvěru k prezidentskému úřadu a vytvořit transparentní
trust to presidential office and create transparent
 kancelář, kde je na prvním místě zájem občanů,“
office where is on first place interest citizens
 uvedl prezident.
said president
 ‘ “As a legal advisor, I believe he will help us restore confidence in the
 presidential office and create a transparent office, where the interests of
 the citizens come first,” the president said.’ (Novinky.cz⁴)

- (4) «Во всех домах есть деревянные дощечки или палки,
Vo vseh domah est' derevjannye doshhechki ili palki
 покрытые какими-то иероглифическими знаками», — отметил
pokrytye kakimi-to ieroglificheskimi znakami otmetil
 он в своем отчете.
on v svoem otchete
 ‘ “In all the houses, there are wooden planks or sticks covered with some
 kind of hieroglyphic symbols,” he noted in his report.’ (trv-science.ru⁵)

- (5) »Erpressung ist das ja sowieso nicht, weil die kein
Blackmail is this anyway not because they no
 Geld wollen«, sagte sie.”
money want said she
 ‘ “It’s not blackmail anyway because they don’t want money,” she said.’
 (Der Spiegel⁶)

Sentences with direct speech are the most straightforward sentences containing quotations. The text in quotation marks is content, and all other components are usually clear to define (see Example (1)). However, there can be various configurations of sentences. Journalists participating in the 2021 JournalismAI

¹http://www.gramota.ru/class/coach/punct/45_192

²https://dictionary.cambridge.org/grammar/british-grammar/reported-speech_2

³<https://prirucka.ujc.cas.cz/?id=162&dotaz=citace>

⁴<https://www.novinky.cz/clanek/domaci-za-pavlem-na-hrad-zamiril-pravnik-pan-ek-40425646>

⁵<https://trv-science.ru/2023/01/rongorongo/>

⁶<https://www.spiegel.de/panorama/gesellschaft/henriette-reker-oberbuergermeisterin-von-koeln-lehnt-deal-mit-letzter-generation-ab-a-516d322e-700e-4541-886a-fd8e9291078d>

Collab Challenges (Guardian [2021]) presented 15 examples in their annotation guidelines.

Despite being the most obvious form for the quoted text, direct speech can be confused with a part of a dialogue in English since the punctuation is the same. In Russian and Czech, the punctuation in dialogues is different (see Example (6)). Dialogue is a distinct phenomenon in language, which is not the subject of my work.

- (6) - Да, я вас слушаю, - сказал он, продолжая писать.
Da ya vas slushayu skazal on prodolzhaia pisat

“‘Yes, I listen to you”, he said continuing writing.’ (2003Anketa.xml_22)

The three above-mentioned components create a cited quote; however, certain elements may be excluded in certain cases. For example, it is possible that the source is not present in the sentence. One of the features of Czech is that personal pronouns can be omitted since the verb is conjugated and carries the grammatical information of a subject (see Example (7)). Therefore there can be no source element in the sentence with a direct speech in Czech, when a name of the author was mentioned earlier in the text.

- (7) “Kdyby se tak mělo stát, potom je otázka,
If REFL so should happen then is question
 zda slovenská strana nadále chce celní unii
whether Slovak side continues wants customs union
 nebo zónu volného obchodu,” **poznámenal.**
or area free trade he.noted

‘ “If this were to happen, then the question is whether the Slovak side still wants a customs union or a free trade area,” he noted.’

(1nd94101-032-p1s3B)

Indirect quotations differ from direct ones in that a sentence with indirect speech focuses more on the content of the reported information than on the exact words. The sentence with indirect speech already contains some analysis of the reported words. It can be done through the choice of reporting verb, adverbs related to the verb, or how the exact words are rephrased.

Indirect quotes could be very different in their structure. In order to computationally extract indirect quotes, I needed to understand their structure better. It was also a necessary step while annotating Czech and Russian sentences. I distinguished four categories (answers to questions, sayings, mixed quotations, and nested quotations) and two categories of structures resembling quotations but in reality they are not (conditions and negated reported verbs).

Answers to questions

Often, the statement identified as a quote is a description of what was said or written, rather than the semantic content (see Example (8)). For instance, it can be the case if a quote is an answer to some question as in Example (9). However, there are cases where the question is needed to understand the quoted answer. In an extreme case, the whole content part is reduced to “it” or “this”, as in

Example (10). In a paper by Newell et al. [2018b], they call such content an *empty content*. Also Pareti [2015] uses a term *empty attribution* for these cases.

- (8) John said a long sentence, answering the question.
- (9) The coach answered the journalist that a defeat is not possible.
- (10) **Polák** *to řekl na včerejší tiskové konferenci v*
Polák this said at yesterday's press conference in
Praze.
Prague
 ‘Polák said this at a press conference in Prague yesterday.’
 (1n94210-57-p2s3)

Sayings

Different sayings and idioms attributed to some group of people can also be recognized as quotations (see Example (11)). However, in my research focusing on media texts, such cases are irrelevant.

- (11) Как говорят азербайджанцы, “у нас в Баку один
Kak govornjat azerbajdzhancy u nas v Baku odin
 армянин трамвай водил, и ничего”.
armjanin tramvaj vodil i nichego
 ‘As the Azerbaijanis say, “we had an Armenian driving a tram in Baku,
 and nothing happened”.’
 (2003Nelegalnaya_perepis.xml_48)

Mixed quotations

Mixed quotations are indirect quotes with directly quoted words. People rarely speak in a way that a full sentence can be taken as a quotation. Using mixed quotations allows one to deliver a coherent message while still keeping some actually said phrases.

- (12) Refalo explained that new subsidiary legislation will help value agricultural plots so that farmers who rent their land can do so at a “fair and appropriate price”.
 (Times of Malta⁷)

Nested quotations

Nested quotations present a quotation of a speaker who quoted some other person. This quotation type can be challenging to annotate and extract computationally since two sources and respective cues should be found.

- (13) **Esso** *said the fields were developed after the Australian government decided in 1987 to make the first 30 million barrels from new fields free of excise tax.*
 Nested quote: **the Australian government** *decided to make the first 30 million barrels from new fields free of excise tax*
 (wsj_0024)

⁷<https://timesofmalta.com/articles/view/agricultural-land-see-new-fair-appropriate-valuation-mechanism.1019404>

Conditions

One example of false quotations is conditional phrases: if something is said under some circumstances, but it is not said in reality (see Example (14)). However, in the text, it looks like a valid quotation and it is likely that the model will classify it as a false positive.

- (14) A pokud FNM řekne ne, vláda s tím
And if FNM says no government with it
nepohne.
not make progress
‘And if the FNM says no, the government will not be able to make any
progress on this.’ (1n94210-57-p2s3)

Negated reporting verbs

This group is similar to the previous one, but there can be two options. The first is that the presence of negation means that words were not said, and then there is nothing to quote. The second option is that negating the reporting verb means negating the reported message’s content as in the Example (15).

- (15) Однако ученые пока не склонны утверждать, что
Odnako uchenye пока ne sklonny utverzhdat', chto
появление термальной аномалии связано с влиянием
pojavlenie termal'noj anomalii svjazano s vlijaniem
радиации и прочих последствий ядерных испытаний.
radiacii i prochih posledstvij jadernyh ispytaniy.
‘However, scientists are not yet inclined to say that the appearance of the
thermal anomaly is due to the effects of radiation and other consequences
of nuclear testing.’ (20030bratnaya_reaktsiya.xml_29)

1.2 Quotations Extraction and Attribution Tasks

In this work, I follow the definitions of quotation extraction and quotation attribution tasks as established by Zhang and Liu [2021]. The quotation extraction task aims to identify a quotation itself, its content, in a document. The quotation attribution task is to connect the content span with the quotation’s author. By the phrase *attributed quotation*, I mean a quoted text together with a source and a cue that establishes the connection between parts.

Quotation extraction can be viewed as a token-based classification task with classes “in quote” or “outside of quote”. But this task can also be understood as a sequence labeling task because a quotation is a connected sequence of words.

The other important point is that quotation is one type of attribution relations. Attribution relation relates an abstract object to an entity [Pareti, 2015]. A quotation shows a relation between an abstract object such as text and an entity e.g. company, authorities, people. The important detail is that the object

was pronounced or written by the speaker in the case of quotation. For example, a belief can be called an attribution relation (see Example (16)), but it is not a quotation.

- (16) **No one in his right mind** actually believes *that we all have an equal academic potential.* (wsj_1286)

Quotations are a subset of attribution relations. So if a system correctly identifies attribution relations, it should also identify quotations.

1.3 Previous Research

1.3.1 Studies on English News Texts

The first work on the automatic detection of direct quotations is paper by Pouliquen et al. [2007]. They created a system named NewsExplorer that analyzed more than 20,000 articles per day in 11 languages. They used the Joint Research Centre’s Europe Media Monitor system as a data source. As mentioned above, Pouliquen et al. introduced the idea of identifying three parts of a quote: “the speaker name, a reporting verb, and the quotation” (source, cue and content in our terminology). Rule-based methods were used for the extraction of reporting verbs and quotations. To detect speakers’ names, they used their own database with 50,000 person names and their variants.

The next important work on the quotation extraction task is by Krestel et al. [2008]. They tried to detect both direct and indirect quotations in the Wall Street Journal dataset. They developed two modules: Reporting Verb Marker and Reported Speech Finder. Both modules used JAPE grammar. JAPE is a tool that works with annotations using regular expressions and functions as a finite-state transducer.

Sarmiento and Nunes [2009] in their work on Portuguese data introduced another approach. They used 19 patterns to extract quotations. The authors noted that “about 5% of the news feeds match these patterns” and that they could not extract quotes with different structures yet with their system. They also went further and trained a Supporting Vector Machine model to get topics from extracted quotes.

Another work on Portuguese data is by Paulo Ducca Fernandes [2012]. Compared to the previously described works, they utilized machine learning methods. The authors worked on two subtasks: quotation identification and quotation attribution. They used the Entropy Guided Transformation Learning (ETL) algorithm for each task. The baseline system used regular expressions and the ETL system outperformed the baseline for each subtask.

In the research of quote attribution, a significant advancement was made with the article by O’Keefe et al. [2012]. The authors reformulated the quote attribution task as a sequence labeling task. They experimented with both binary and n-way class models in their work and with three sequence decoding models: greedy, Viterbi, and a linear chain Conditional Random Fields (CRF). Their approach focused solely on direct quotes and the direct portion of mixed quotes. To extract quotes, they employed regular expressions, achieving 99% accuracy for a clean English-language dataset. The feature set for quotation attribution

was based on the work of Elson and McKeown [2010]. The Viterbi decoders demonstrated stable results on the WSJ corpus, with more than 83% accuracy for quotation attribution for both binary and n-way class models.

One of the biggest problems for quotation attribution is coreference resolution and anaphora resolution as one of its varieties. Almeida et al. [2014] devised a model of a quotation-coreference tree that solved these two problems jointly.

Newell et al. [2018a] adopted the approach suggested by Pareti et al. [2013] and enhanced it. The system can be divided into three parts: verb-cue classifier, quote content classifier, and quote source classifier. The first classifier used CNN, and two later classifiers used CRF. The enhancements were the content resolver and the source resolver. For both of them, they trained Max Entropy classifiers.

Pavlo et al. [2018] presented an unusual approach for quotes extraction and attribution named Quootstrap. They modified pattern-based quotation attribution by adding bootstrapping. First, they used a seed pattern, which is the simplest quotation pattern for instance [Quotation said Speaker]. After finding all entries that match the pattern, the algorithm searches for found Quotations and Speaker pairs. Then in newly found sentences with these words, a new pattern is added based on the sentence content. The precision of the model is relatively high, but recall is low.

Vaucher et al. [2021] described the development of Quootstrap: a framework called Quobert. In the same paper, they presented a dataset Quotebank that was created using Quobert (details on Quotebank 1.4). The authors call their approach distantly supervised because it combines supervised and unsupervised methods. Quobert leverages the bootstrapping principle to generate training data with minimal supervision, which is then utilized for training a supervised model to improve overall performance in general and recall in particular. This approach avoids the necessity of manually labeled input. Instead, it exploits the redundancy of the corpus by bootstrapping from a single seed pattern to extract training data for fine-tuning a BERT-based model. Quobert is claimed to be agnostic to the fact that the quote is direct or indirect, but the authors used it only on direct quotes. Using the quotation extracted with Quobert, they fine-tuned BERT for the quotation extraction task.

1.3.2 Studies on Literary Texts and Other Languages

In parallel with work on news data, researchers started to look at the extraction and attribution of quotes in literary texts. Although the task is the same, the nature of quotations in these texts differs. Elson and McKeown [2010] worked on automatic attribution of quoted speech in a corpus of novels written in the 19th and 20th centuries, with more than 3000 quotations. All named entities and nominals appearing in the text preceding the quote were considered candidate speakers. Then each quote was classified into one of the predefined 19 syntactic categories. For each candidate-quote pair, the authors calculated a vector of quantitative features such as, for instance, the distance between the candidate and the quote, the number of appearances of the candidate, and the length of the quote. The achieved accuracy was 83%.

O’Keefe et al. [2012] tested the system, which primarily oriented on news texts, on literary text and could not outperform the baseline system, which proves that

texts from different domains should be handled differently. However, ideas used for one type of texts were adapted and tried on another. Muzny et al. [2017] proposed an idea of a two-stage sieve approach for quote attribution and tested it on novels. First, they connected the quotation with mentions of actors who could have said it. Second, they linked mentions with the actors, entities. Mentions were pronouns or other words that could be used to refer to some entity. Entities were characters of novels.

One of the most recent works on quotation extraction and attribution relates to literary texts. Cuesta-Lazaro et al. [2022] used BERT and Dialogue State Tracking techniques to detect utterances and characters and attribution in novels.

Along with already mentioned studies on Portuguese, researcher work on systems for other languages. There are studies on Estonian [Särg et al., 2021] and Norwegian [Salway et al., 2017], and a framework for Indonesian was proposed [Purnomo W.P. et al., 2021].

1.4 Datasets

I explored different corpora that have annotation for quotation extraction and attribution. In this section I provide an overview of the English datasets that I was considering for training and one Czech dataset that I used for additional evaluation.

PARC

PARC was created for research on attribution relations in the first place. As I write in section 1.2, quotations belong to the class of attributions, and their extraction is the same in many aspects; that is why PARC suits the research on quotation extraction.

The authors of PARC used Penn Discourse Tree Bank (PDTB) as a starting point. The PDTB is a corpus of discourse relations built on the Wall Street Journal section of the Penn Treebank [Prasad et al., 2008]. The PDTB includes annotations for attribution relations in addition to other discourse relation types.

While building an annotation scheme for the attribution task, Pareti [2016] utilized the PDTB annotation scheme as a foundation but modified it to fit the task’s requirements better. The attribution relations annotated in the PDTB are kept in PARC with corresponding labels.

Each attribution relation consists of source, cue, and content spans. There is also an optional supplement span which is “any additional element relevant to the interpretation of the attribution relation, such as expressing information” [Pareti, 2016].

PolNeAR

Political News Attribution Relations Corpus 2016 or PolNeAR was presented by Newell et al. [2018b]. As a data source, they used articles from seven publishers representing different political views. Additionally, they analyzed the PARC dataset and reported its limitations.

DirectQuote

The DirectQuote⁸ corpus was designed primarily for the quote extraction task. The authors used texts from 13 popular online new media from several English-speaking countries [Zhang and Liu, 2021]. Overall 10,279 quotations and the corresponding speakers were manually annotated. The raw sentences are tokenized by a whitespace tokenizer, and each token is labeled. The dataset does not provide any other information: only tokens annotated with quote/speaker labels.

Quotebank

Quotebank is the most recent English corpus; it was presented in 2021. It contains 178 million attributed quotations. The corpus is automatically labeled. In order to create such a large corpus, [Vaucher et al., 2021] developed a framework named Quobert (see Section 1.3).

The authors applied Quobert to 162 million news articles obtained from Spinn3r – a content aggregation service [Vaucher et al., 2021]. The Quobert framework was mainly developed for detecting direct quotations and the Quotebank contains only them.

SiR1.0

The Sources in iRozhlas 1.0 (SiR1.0) dataset consists of annotated newspaper articles, which were annotated through a crowdsourcing effort [Hladka et al., 2022]. This task involved around 2,000 articles and over 290 annotators who identified and marked over 11,000 citation cues (signals in their terms) and approximately 10,000 citation sources, along with their types and connections to the cues. The corpus is divided into three parts depending on how many annotators worked on it.

Authors of SiR annotated only sources and cues since their focus was on attribution rather extraction of quotations. Since there is no annotation of content spans, this dataset can not be used for proper evaluation of the models, but I use it for additional evaluation.

1.5 Metrics

There are no standard metrics to measure the quality of quote extraction and its attribution. However, three types of metrics are usually used in the research: precision (P), recall (R), and F-measure (F1).

Precision, recall, and F-measure in general

The precision metric evaluates a model’s capacity to avoid assigning a positive label to a sample that is actually negative. Formally, precision is defined by the formula 1.17, where $\#TruePositives$ is the number of examples whose predictions

⁸<https://github.com/THUNLP-MT/DirectQuote>

Dataset	Source	Release year	Size
PARC	PDTB (Wall Street Journal)	2016	$\sim 19,712$ attributions
PolNeAR	<ul style="list-style-type: none"> • New York Times • Washington Post • USA Today • Breitbart • Politico • Huffington Post • Western Journalism <p>with the focus on the 2016 US President Elections</p>	2018	24,000 attributions
Quotebank	English-language news articles from Spinn3r.com from the period August 2008 to April 2020	2021	178 million attributions
DirectQuote	13 online news media from five major English-speaking countries	2021	10,353 direct quotations
SiR 1.0	The Czech public radio iRozhlas	2022	10,110 attributions

Table 1.1: Datasets statistics

match gold labels, $\#FalsePositives$ is the number of examples that were wrongly assigned with a positive label.

$$P = \frac{\#TruePositives}{\#TruePositives + \#FalsePositives} \quad (1.17)$$

Recall, also known as sensitivity or true positive rate, measures the ability of a model to correctly identify and capture all the relevant positive instances from the entire set of actual positive instances. Recall is calculated with the formula 1.18, where $\#FalseNegatives$ are elements that were wrongly classified with negative labels.

$$R = \frac{\#TruePositives}{\#TruePositives + \#FalseNegatives} \quad (1.18)$$

F1-score is a harmonic mean of precision and recall, and it shows how balanced the model is.

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (1.19)$$

Precision, recall, and F-measure in quote extraction

In quotation extraction, we can evaluate a model on individual tokens or on whole quotes.

When considering tokens, true positives are tokens that were correctly predicted with a label; false positives are tokens that do not belong to a quotation, but the model labels them as part of a quotation. Precision assesses the model’s proficiency in accurately identifying quote candidates’ tokens that are, in fact, genuine quotations. Recall refers to the model’s ability to recognize tokens accurately that are part of a quotation but may have been missed or not identified by the model.

Pareti [2015] operated with three metrics approaches: strict, partial, and soft. The strict approach considers a span correct only if it exactly matches the annotation. In the partial scheme, the proportion of predictions and gold labels overlap is used as $\#TruePositives$. For the soft approach, any overlap of predictions and gold labels is counted as a true positive.

In the Quootstrap system, the precision of the extracted quote is ensured through the algorithm [Pavlo et al., 2018]. The identified quotation and the speaker are connected together and considered as a pair. Precision and recall for this configuration are defined through the sets of ground truth and predicted pairs of elements. A pair is counted as correct if the quotation was identified and it was attributed to the right speaker.

1.6 Description of CRFs

Conditional Random Fields (CRFs) is a statistical modeling technique that aims to predict sequences, such as named entities or quotes, based on the likelihood of observing a specific sequence of states. Introduced by Lafferty et al. [2001], CRFs are random fields conditioned on input data, with the joint distribution of

the label sequence Y given the input X . The formal definition of a conditional random field is given in

Definition. Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G . Then (X, Y) is a conditional random field in case, when conditioned on X , the random variables Y_v obey the Markov property with respect to the graph: $p(Y_v | X, Y_w, w \neq v) = p(Y_v | X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G .

A random field is a generalization of a stochastic process over a multidimensional space in which the “time” parameter can be a vector instead of a single integer. When conditioned on input X , the random variables Y_v in a CRF obey the Markov property with respect to the graph G .

Linear-chain CRFs, a specific type of CRF, operate with feature functions that depend only on the current and previous labels. These functions take as input a sentence, the position of a word in a sentence, and the labels of the current and previous words. Feature functions often take the form of indicator functions, providing binary values that indicate the presence or absence of specific features. The weights associated with these functions represent the strength of the association between the given feature and label.

CRFs are often compared to Hidden Markov Models (HMMs). The key difference between the two is that CRFs are discriminative models, while HMMs are generative ones. This distinction enables CRFs to capture complex dependencies through feature functions. In Figure 1.1, I compare the structures of HMM, linear-chain CRFs, and general CRFs.

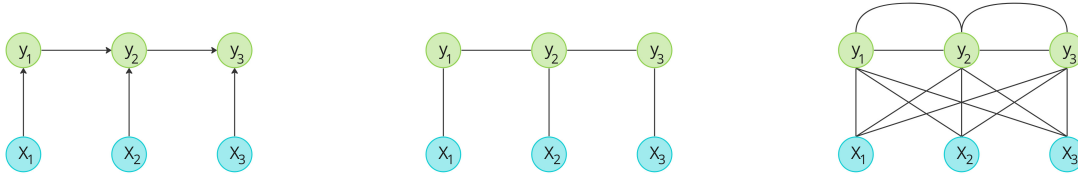


Figure 1.1: HMM, linear-chain CRFs, general CRFs

CRFs have become a popular choice for Named Entity Recognition tasks, and they have also been applied to quotation extraction tasks, which can be viewed as sequence labeling tasks [O’Keefe et al., 2012]. The effectiveness of CRFs in these applications can be attributed to their ability to model complex relationships and dependencies among words, labels, and features in a given sequence.

After examining the theoretical aspects, I can proceed to discuss the data and techniques employed in developing a system that detects quotations in not just the language it was trained on, but also across other languages.

2. Experiments

In this study, my objective is to create a language-independent system capable of automatically identifying direct and indirect quotations within news articles. This system is compared with a baseline model. In this chapter, I start with the description of the training dataset and annotation process of Czech and Russian data. Then I define the evaluation metrics that I used and explain the baseline and machine learning-based systems in detail.

2.1 Datasets

One of the objectives of this work is to develop a system that can be used with other languages. I selected Czech and Russian for testing the approach due to my association with Charles University and Russian being my native language. To my knowledge, there are no quotation attribution systems and corpora with annotated quotations and attributions for these languages, so I annotated small datasets myself, relying on the Annotation guide created for the PARC dataset (Pareti [2016]). In the following sections, I describe the dataset for each language and the annotation process in detail.

2.1.1 English Data

I chose to work with data for English primarily because of the availability of datasets. However, as I described in Section 1.4, there are not so many corpora that have annotations for quotations. Here, I describe the chosen dataset for English in detail.

As the primary corpus, I chose the Penn Attribution Relations Corpus (PARC 3.0). This corpus is specifically dedicated to attribution relations, but it is still used for quotation extraction research [Newell et al., 2018a], [Newell et al., 2018b]. Moreover, the system trained on the more general phenomenon, such as attribution relation, should work on a subset, such as quotation. PARC 1.0 was released in 2012 Pareti [2012a]. The final version of PARC 3.0 was released together with the paper by Pareti [2016]. The dataset is not available publicly, however, it can be obtained from its author Silvia Pareti.

PARC is based on the Penn Discourse Treebank (PDTB) and keeps its annotation of attribution relations. Sometimes, PDTB annotation overlaps with PARC annotation, and in these cases, I give priority to the largest annotation span in terms of tokens.

PARC contains 19,712 annotated attribution relations and 48,427 sentences. The annotation schema comprises *source*, *cue*, and *content* tags. The tag *source* is for a person or an entity to which the quote is attributed. The verb or some type of a hint which tells the reader that the following text is a quote is annotated with a *cue* tag. The *content* tag is for the quoted content itself.

The general inter-annotator agreement for attribution relations identification is 0.79; however, the inter-annotator agreement for each element of quotations is more than 0.90 [Pareti, 2016].

The authors of the PARC dataset annotated separately nested attributions (see Example (13)). I excluded them from my research and worked only with top-level cases.

The average cue length in the PARC training dataset is 1.45 tokens. There are 4050 sentences (27% of all annotated sentences) with cues longer than 1 token.

I validated the dataset using raw PDTB files and found some missing symbols in the PARC dataset. For example, in 16 documents from the test dataset (folder 23 of PDTB), the final quotation mark in the document was missing. It was also the case for three documents in the development set (folder 24) and for 192 documents training dataset (folders 00-22 of PDTS). I manually added missing quotation mark tokens in the test part of PARC because it was critical for baseline testing.

Sometimes clear quotations are not annotated in the dataset as in Example (1).

- (1) “The morbidity rate is a striking finding among those of us who study asbestos-related diseases,” said Dr. Talcott. (wsj_003)

The dataset comprises a wide variety of quotations in terms of size. For instance, the median cue length is one token, but there are outliers, such as the sentence in Example (2). The cue there has a length of 13 tokens. This annotation was inherited from the PDTB dataset.

- (2) **Executives at Olivetti, whose earnings** have been steadily sliding over the past couple of years, have acknowledged *that in the past, they have lagged at getting new technology to market.* (wsj_1591)

The longest content span in the PARC training set is 78 tokens (see Example (3)).

- (3) **The labels** were breathy: *“Within its sheltering walls is a microcosm of a thousand years in garden design ... a rose garden, herb garden, serpentine garden, flower fields, an apple orchard ... organized in a patchwork of 50-by-50-foot squares to form ‘rooms’ ... here and there are simple architectural forms, a whimsical jet of water, a conceit of topiary or tartan plaid, and chairs of every sort to drag around* (wsj_0984)

These examples show that this dataset is suitable for benchmarking the quote attributions system since it contains sentences of various types.

There are some cases of incorrect annotation. For example, what is annotated as a cue (21 tokens) in Example (4) should be annotated as a source instead.

- (4) *“We thought it was awfully expensive,”* **said** Sterling Pratt, wine director at Schaefer’s in Skokie, Ill., one of the top stores in suburban Chicago *“but there are people out there with very different opinions of value.* (wsj_0071)

2.1.2 Czech Data

I annotated a portion of the Prague Dependency Treebank (PDT) for the Czech test dataset. The corpus consists primarily of news, as well as business and popular scientific articles from the 1990s. The treebank contains 87,913 sentences and about 1.5 million tokens. I used the version of the PDT presented on the Universal Dependencies page, the Czech-PDT UD treebank, since the corpus there is in the CoNLL-U format Nivre et al. [2016], which is sufficient for my purposes and is easier to process than the XML format of the full PDT corpus.

I used the following algorithm for annotation.

1. Select sentences with quotation marks;
2. Select sentences with reporting verbs from the portion of sentences without quotation marks;
3. Randomly order sentences with quotation marks;
4. Randomly order sentences without quotation marks selected in step 2;
5. Translate sentences into Russian;
6. Annotate 42 sentences with quotation marks and 66 sentences with a reporting verb but without quotation marks;
7. Add 108 random sentences from the group of sentences without quotation marks and reporting verbs.

Now, I describe the procedure in more detail.

For my dataset, I wanted to include sentences containing direct, indirect, and no quotes. Therefore, I chose 42 sentences with quotation marks, 66 sentences with a reporting verb but without quotation marks, and 108 sentences containing neither a reporting verb nor quotation marks. The sentences were chosen randomly but roughly equally from each of the four parts of the corpus, where each part reflects a different source of the dataset. I manually annotated all these sentences.

I automatically translated the sentences from Czech into Russian, and together with my knowledge of Czech, it sufficed for the annotating process. I consulted my supervisor in some complicated cases. I annotated tokens with labels **source**, **cue**, **content**, and **None**, so it is compatible with the English dataset.

The annotation was done on separate sentences, and all quotes longer than one sentence were ignored. The resulting dataset is a collection of sentences, not texts.

During annotation, I intentionally skipped sentences where the quoted message was represented only with a pronoun or cases when the quoted text was conditional or negated (see Section 1.1). Also, I did not annotate parts of the dialogues.

In the case of direct quotes, I labeled as content all tokens inside quotation marks, including quotation marks themselves. This is the approach that was used in PARC.

The longest quote’s content contains 43 tokens, and this sentence is Example (5).

- (5) Jak sdělil, celá měnová krize, která v poslední době
 As he.said, whole passed crisis, that at last time
 v ES nastala, je signálem toho, že násilný pokus
 in EC occurred, is signal of that violent attempt
 o vytvoření jedné měny nemůže mít v nejbližší
 to create one currency cannot have in near
 budoucnosti sebemenší šanci na úspěch, protože
 future slightest chance on success, because
 ekonomická situace v jednotlivých zemích ES je
 economic situation in individual countries EC is
 značně rozdílná.
 very different.

‘As he said, the whole currency crisis that has recently occurred in the EC is a signal that a violent attempt to create a single currency cannot have the slightest chance of success in the near future, because the economic situation in each EC country is very different.’ (mf920922-079-p2s3)

2.1.3 Russian Data

As the primary data source, I used the Universal Dependencies version of the SynTagRus corpus (Droganova et al. [2018]). This dataset consists of 25,447 sentences and 409k words.

The annotation procedure for Russian is slightly different but practically the same as for Czech.

1. Select sentences with `journalism` tag from the file `train-c` and with reporting verbs;
2. Select sentences with quotation marks from files `train-a` and `train-b`;
3. Randomly order sentences with quotation marks;
4. Randomly order sentences selected in step 1;
5. Annotate 56 sentences with quotation marks;
6. Annotate 44 sentences without quotation marks from the portion selected in step 1;
7. Add 108 random sentences from the file `train-c`, that do not have quotation marks or reporting verbs.

Next, I will elaborate on the algorithm in greater detail.

Sentences in the most recent file of the dataset have labels displaying the text genre: journalism, fiction, etc. I filtered those sentences that had label `journalism` and from them selected ones that had reporting verbs in them. From

the other files, I selected sentences that had quotation marks. Similar to Czech, the resulting dataset contains 100 sentences with direct or indirect quotes and 108 sentences without quotes.

Although Pareti [2016] did not distinguish between dialogue lines and direct speech for the PARC dataset, I did not include dialogue lines while annotating Russian sentences. Dialogues in Russian employ different punctuation compared to direct quotations (refer to Section 1.1), so including sentences with dialogue lines would have resulted in inconsistencies when compared with other datasets.

Same as for the annotation of Czech, I worked on a sentence level and did not include quotations that extended to more than one sentence.

In the standardization process, I noticed some tokens in the dataset that are not present in the sentence. I encountered a case of the omitted predicate, where the implied token was represented by a dash in a sentence in Russian according to language grammar conventions but was presumably manually added into a CoNLL representation of a sentence. I removed such tokens because I needed only those tokens that were in the sentences.

The longest content span is 61 tokens (see Example (6)).

- (6) Как разъясняет **следователь**, “*основная содержательная*
Kak razjasnjaet sledovatel’, “*osnovnaja soderzhatel’naja*
нагрузка и цель данных экспонатов состоят в том,
nagruzka i cel’ dannyh jeksponatov sostojat v tom,
чтобы транслировать следующие идеи-утверждения: что
chtoby translirovat’ sledujushhie idei-utverzhdenija: chto
равноценны и равнозначны (сопоставимы) образы Иисуса
ravnocenny i ravnoznachny (sopostavimy) obrazu Isusa
Христа и Микки Мауса; что равноценны и
Hrista i Mikki Mause; chto ravnocenny i
равнозначны (сопоставимы) по своему культурному и
ravnoznachny (sopostavimy) po svoemu kul’turnomu i
нравственному содержанию православное христианство и
nravstvennomu soderzhaniju pravoslavnoe hristianstvo i
любой медийный продукт, например, мультфильм про
ljuboj medijnyj produkt, naprimer, mul’tfil’m pro
Микки Мауса...”.
Mikki Mause...”.

‘As the investigator explains, "the main content load and purpose of these exhibits is to transmit the following ideas-assertions: that the images of Jesus Christ and Mickey Mouse are equivalent and comparable; that Orthodox Christianity and any media product, such as a cartoon about Mickey Mouse, are equivalent and comparable in their cultural and moral content..."’
(2009Zapretnoe_iskusstvo_2006.xml_46)

Overview of the Datasets

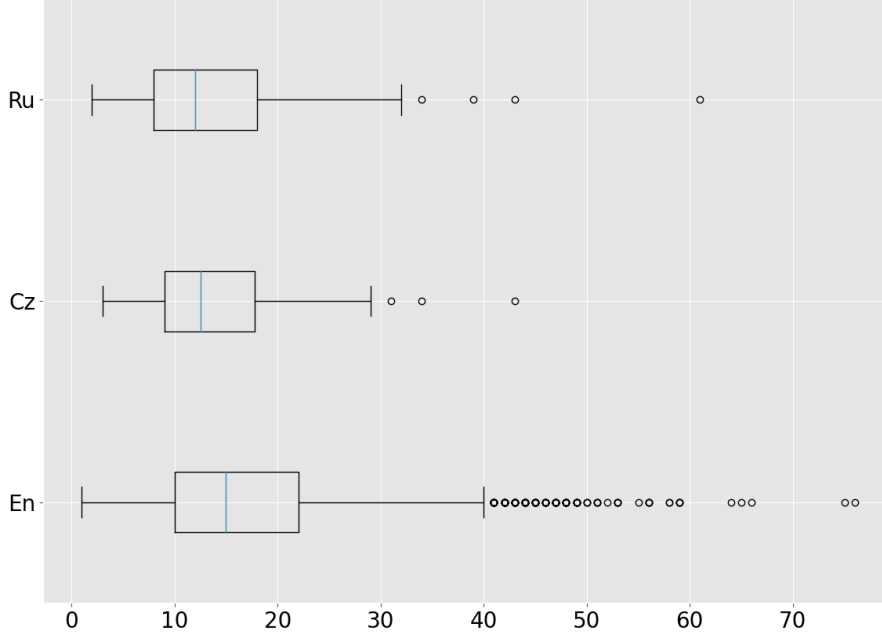


Figure 2.1: Length of content spans in three datasets

In Figure 2.1, I depicted the statistics on the length of content spans in all three datasets. In the English dataset, there are quotations that are longer than one sentence, but I excluded them from these statistics. The median content span length in the English corpus is 15 tokens; in Russian and Czech corpora, it is smaller, 12 and 12.5 tokens, respectively. The discrepancies mainly stem from the differences in dataset sizes. However, the statistics remain comparable, making the Czech and Russian datasets suitable for experiments’ evaluation.

2.2 Standardization of Datasets

The PARC dataset is available in XML format. Each word has several annotation tags, including ones related to attributions if it is present in the sentence. It required several steps to process the data. Here is an example of the dataset structure in Figure 2.2.

Inside the attribution tag, there is an id of attribution. These ids show the connection between the source, cue, and content parts. Since the annotation was based on different sources, such as available PDTB annotation and efforts of Pareti’s team, it is also indicated in the label.

The corpus format allows a single word to belong to multiple attributions. There are two reasons for this: one is to capture nested attributions, and the second is to annotate words belonging to both cue and source. An example of

```

[...]
<S gorn="15">
  <SBAR-ADV gorn="15,0">
    [...]
    <VP gorn="15,0,1,0,1">
      <WORD ByteCount="1360,1366" gorn="15,0,1,0,1,0"
        lemma="admit" pos="VBZ" sentenceWord="3"
        text="admits" word="245">
        <attribution
          id="wsj_1122_Attribution_relation_level.xml_set_15">
            <attributionRole roleValue="cue"/>
          </attribution>
        </WORD>
      <NP gorn="15,0,1,0,1,1">
        <WORD ByteCount="1367,1370" gorn="15,0,1,0,1,1,0"
          lemma="it" pos="PRP-S" sentenceWord="4"
          text="its" word="246">
          <attribution
            id="wsj_1122_Attribution_relation_level.xml_set_15">
              <attributionRole roleValue="source"/>
              <attributionRole roleValue="content"/>
            </attribution>
          </WORD>
        <WORD ByteCount="1371,1376" gorn="15,0,1,0,1,1,1"
          lemma="error" pos="NN" sentenceWord="5"
          text="error" word="247">
          <attribution
            id="wsj_1122_Attribution_relation_level.xml_set_15">
              <attributionRole roleValue="content"/>
            </attribution>
          </WORD>
        </NP>
      </VP>
    </S>
  [...]

```

Figure 2.2: Example of markup in PARC

nested attribution is in (7). Nested attribution is: **it** [...] is considering *appealing Judge Curry’s order*. All tokens in the nested attribution have two attribution ids and can have different labels. For example, *is considering* is both content and cue. In this thesis, I ignored nested quotes.

- (7) **Commonwealth Edison** said *it is already appealing the underlying commission order and is considering appealing Judge Curry’s order*. (wsj_0015)

In the training set of PARC, there are 50 sentences with words with two attribution roles. Overall, there are 111 words with two attribution roles. Most of the tokens with multiple tags belong to nested attributions (S. Pareti, personal communication, November 11, 2022). But there are also cases of possessive expressions, when source and cue are represented by the same token or tokens. This is the situation of the word *its* in the Example (8). Since I store such labels separately, meaning that if a token has two labels, then this token is included in both lists, such cases do not influence the model.

I needed single labels for words, so in the situation of two labels, I prioritized cue and **source** over *content*, and I considered cue label more important than **source** since verb cues play a big role in the extraction process.

- (8) If **the IRS** admits *its* error and the charges have been paid, it will reimburse a taxpayer who hasn’t refused to give timely answers to IRS inquiries or hasn’t contributed to continuing or compounding the error. (wsj_1122)

Each token has **ByteCount** attribute, which contains starting and ending bytes of a token. I used this attribute to form full sentences from PARC’s data. I had full sentences available in Czech and Russian datasets, so I used them as input where required.

In PARC, round brackets were replaced with **-LRB-** and **-RRB-**, and curly brackets with **-LCB-** and **-RCB-**. I replaced these tags with the appropriate brackets symbols. A similar situation was with quotation marks. Double symbols ‘ ‘ and ’ ’ were used instead of one-symbol quotation mark " .

Single quotation marks are also used in PARC, usually for nested quotations. Example (9) shows a case of a nested quotation in single quotation marks. I did not replace single quotation marks with double quotation marks.

- (9) “*In Moscow, they kept asking us things like, ‘Why do you make 15 different corkscrews, when all you need is one good one?’* ” **he** says. (wsj_0102)

2.3 Evaluation Methods

In my study, I employed metrics focusing on two language levels: tokens and sentences. Metrics based on words can show how precise the model is, whether it does not include non-relevant words or mislabel correct words. As metrics based on words, I measured precision, recall, and F1 for each component of the

quotation: source, cue, and content. I used macro-averaging for the evaluation of the overall performance of components.

In an example of one sentence, I show how exactly the method works. In (10), tokens are labeled according to the dataset. If the model made wrong predictions and the result looked like in Example (11), then the metrics for the source would be:

$$\begin{aligned} \text{precision} &= \frac{2}{3} = 0.67 \\ \text{recall} &= \frac{2}{8} = 0.25 \\ F1 &= \frac{2 \cdot \frac{2}{3} \cdot \frac{2}{8}}{\frac{2}{3} + \frac{2}{8}} = \frac{4}{11} = 0.36 \end{aligned}$$

- (10) **Mr. Giuliani’s campaign chairman, Peter Powers, says the Dinkins ad is “deceptive.”** (wsj_0041)
- (11) Mr. Giuliani’s campaign chairman, **Peter Powers, says** the Dinkins ad is “deceptive.” (wsj_0041)

For metrics based on sentences, the detected quotation’s boundaries are unimportant. It is binary classification: if it was predicted that the sentence contains a quote and it actually does, then it is counted as true positive case. In real working systems, for example, for journalistic purposes, it might be enough only to provide a sentence with a quote. For a journalist, how precise the system extracts quote spans might not be that important. That is why I used this approach as well. The prediction as in Example (11) counts as correct, since the content span was found in the sentence.

Another valuable parameter I measured is whether all components belong to one attribution set. PARC provides annotation of full attributions, so I could see if all parts were connected together in an attribution. The attribution was considered correct if the correct words belonged to it. If some wrong word was included in an attribution, then the attribution is incorrect. However, if some words are missing from the attribution, I did not consider it wrong.

For comparability, I also used the approach by Newell et al. [2018a] and considered the span correct if it is matched exactly to the annotated span. This way only *cue* span would be considered correct in Example (10).

2.4 Implementation Details

I used the SpaCy library for all manipulations during my work. SpaCy is an open-source Python library designed for advanced natural language processing (NLP) tasks.¹ It is an efficient and fast performance package, making it suitable for use in real-world applications and large-scale data processing. SpaCy provides capabilities for various NLP tasks, such as part-of-speech tagging, named entity recognition, syntactic dependency parsing, tokenization, and lemmatization. It also supports multiple languages and offers pre-trained models for specific tasks, which can be easily integrated into custom applications.

¹<https://spacy.io/>

SpaCy uses processing pipelines that turn text into a collection of predictions for different tasks. I implemented pipeline components for my experiments: for a baseline model and separate pipelines for each component of the machine learning model.

SpaCy provides language models for English and Russian. The English models were trained on OntoNotes 5 [Weischedel et al., 2022] and WordNet 3.0 [Miller, 1995]. I used the model `en_core_web_sm` in my work. The Russian language model is trained on Nerus dataset.² SpaCy does not provide a language model for Czech, so I used a compatible module `spacy-udpipe` that provides an interface to use the UDPipe model.

UDPipe is an open-source toolkit for natural language processing (NLP) that provides various NLP functionalities such as tokenization, part-of-speech tagging, morphological analysis, dependency parsing, and named entity recognition (Straka [2018]). UDPipe uses a multi-layer feed-forward neural network that takes as input word embeddings and character-level word embeddings. This neural network is trained jointly on multiple tasks, including tokenization, part-of-speech tagging, and dependency parsing. The idea behind multi-task learning is that the neural network can learn to share representations across different tasks, which can help improve performance on all tasks.

UDPipe language models were trained on Universal Dependencies treebanks.³

The common problem that I had with all datasets is that they are already tokenized. For dependency parsing features, I had to provide entire sentences, and the parser sometimes tokenizes them differently than they are tokenized in the dataset. Gold labels depend on tokenization, which is also why it is important. For instance, in (12) compound word *Minneapolis-based* was annotated as one token in PARC, but SpaCy splits it into three tokens: *Minneapolis*, *-*, *based*. It happened with other words with hyphens.

- (12) While many of the risks were anticipated when Minneapolis-based Cray Research first announced the spinoff in May, the strings it attached to the financing hadn't been made public until yesterday. (wsj_0018)

I solved this problem using SpaCy Example functionality and having differently tokenized data tied together.⁴ The architecture of the SpaCy pipeline that I use for training models is illustrated in Figure 2.3. The reference document contains tokens and labels from the datasets; the predicted document keeps tokens processed by SpaCy from the provided sentences and predictions for each token.

The detailed description of the organization of the code is provided in Appendix B.

2.5 Baseline Model

I chose a rule-based model from Textacy⁵ as a base for the baseline model, which I further modified. Its algorithm is loosely based on the paper by Krestel et al.

²<https://github.com/natasha/nerus>

³<https://ufal.mff.cuni.cz/udpipe/2/models>

⁴<https://spacy.io/api/example>

⁵<https://textacy.readthedocs.io/en/latest/index.html>

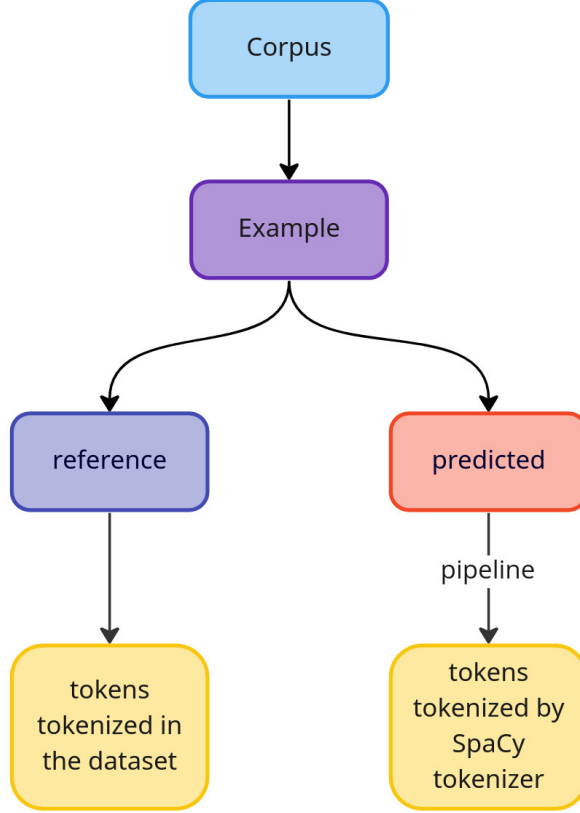


Figure 2.3: Diagram of data relations

[2008]. First, the algorithm checks if there is an even number of quotation marks in the given text. If it is an odd number, then a quotation mark is missing, and its potential location is unclear. If there is an even number of quotation marks, the document is split by the marks. Also, the algorithm filters out quoted segments with less than four words to filter out titles or other names that could be in quotation marks. Next, the function considers parts of sentences outside quotation marks and adjacent sentences to find a cue inside them. It uses a list of reporting verbs. Dependency parsing is used to find the source. If a reporting verb is found, the source is detected as a token with nominal subject or clausal subject dependencies.

This model works only on direct quotations. It uses a list of reporting verbs for cue identification. For English, I used the list of verbs available in the Textacy package. For Czech, I developed the list together with my supervisor, Czech speaker. The Russian list of reporting verbs was built by myself. These lists are presented in the appendix A.1, A.1, A.3.

The model considers both double and single quotation marks. However, it sometimes struggles to differentiate between a single mark used as an apostrophe and one used as a quotation mark. To address this issue, I introduced a condition that the model should only consider double quotation marks. In the PARC dataset, single quotation marks are exclusively used for nested quotes, which are not a focus of my research, making this decision justifiable. Furthermore, both Czech and Russian datasets only utilize double quotation marks for quotations, reinforcing the rationale behind this choice.

2.6 Machine Learning-Based System

I used the idea of Newell et al. [2018a] for the machine learning-based model. The code of their work is not available, so I used only the description from the paper as guidance.

The concept behind the system involves employing three distinct classifiers for each quotation component. Plus, there are two more classifiers called resolvers. The content resolver’s goal is connecting content spans with cues; the source resolver does the same but with sources and cues. The cue is a connecting element for the attribution of quotations. The system is visualized in Figure 2.4. Solid lines mean that the predictions are used for training in the respective components, while the dashed lines mean I used gold labels for training and predictions only for inference. I explain the connections in detail in the following sections dedicated to each component.

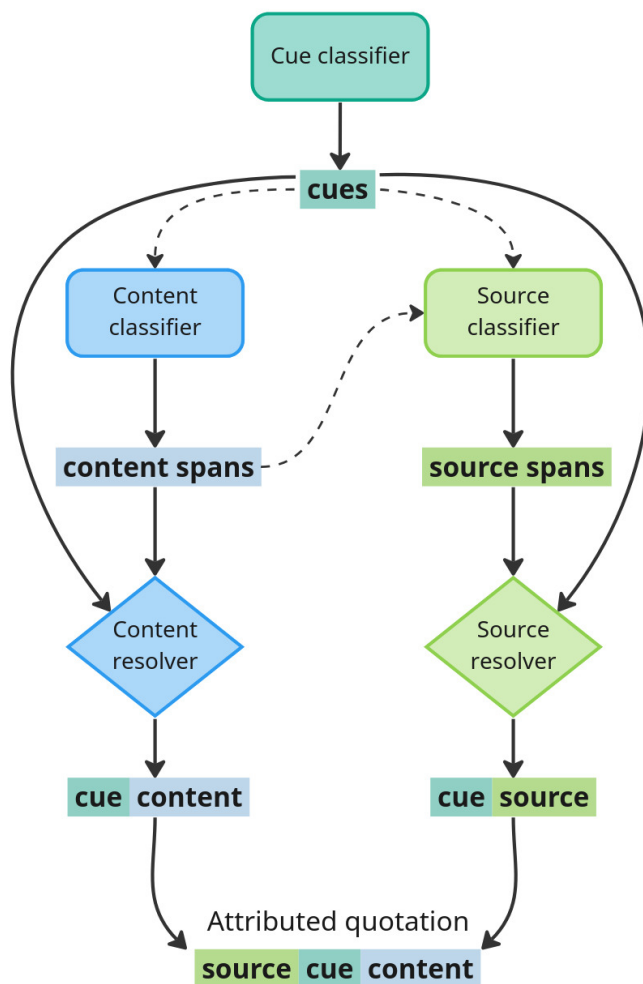


Figure 2.4: Machine learning-based system

Cue Classifier

The essential component of the machine learning-based system is a cue classifier. Its purpose is to find words (usually verbs) that signalize that a quoted text was

produced by a person or an entity mentioned in the text (source). Since the cue connects these two parts, the cue classifier is a critical component of the whole system.

There are different ways to detect cues in the test. The simplest method is to compare tokens with the list of reporting verbs. This approach is used in the baseline model.

The other method is to use a classification model that would classify each token, whether it is a cue or not. Moreover, a cue can be a span of tokens, as in Example (13), so the model should be able to find groups of tokens. Newell et al. [2018a] suggested using SpaCy Named Entity Recognition (NER) model as a cue classifier. This selection is well-suited, as NER models yield token spans accompanied by IOB-labels, such as inside, outside, and beginning.

(13) It has long been rumored *that Ocean Drilling would sell the unit to concentrate on its core oil and gas business.* (wsj_0313)

The aim is to develop a NER model from the ground up, utilizing the PARC annotations of cues. The SpaCy NER model is a Convolutional Neural Network (CNN) model. I built training examples using the tokens labeled with cue tag.

However, it is not possible to use the NER trained on the English corpus for other languages as it is. Doing that would require training NER on a large labeled dataset similar to PARC but for other languages, which does not exist to the best of my knowledge. Building such a dataset from scratch is out of the scope of this work. Therefore, I labeled as cues tokens which lemmas match with the list of reporting verbs lemmas. These lists can be found in Appendix A.2, A.3.

Content Classifier

Following Newell et al. [2018a] approach, I used the Conditional Random Fields model in my experiments as a content classifier. I used a patched version of the `sklearn` wrapper CRFSuite by Okazaki [2007].⁶

As the features for every token, I used:

1. text of the token;
2. token's lemma;
3. previous five tokens
4. following five tokens;
5. flag if the token's sentence contains a cue;
6. flag if the token follows a cue;
7. token's dependency depth;
8. token's dependency relation;

⁶<https://github.com/TeamHG-Memex/sklearn-crfsuite/pull/69>

9. index of token in the sentence;
10. token’s part-of-speech tag from the Universal POS tags;⁷
11. token’s detailed part-of-speech tag;
12. token’s IOB tag;
13. flag if the token is a child of a cue;
14. flag if the token is the leftmost child of a cue;
15. flag if the token is in quotation marks;
16. flag if the token is a quotation mark.

The first two features were straightforward to obtain. The previous and following five tokens were extracted on the document level, so if, for example, the token is in the second position in the document, then it has only one previous token. For these four features, I will use the term *text features* further because they contain tokens from the text or their forms (lemmas).

The following two features are binary features about the presence of the cue in the sentence and whether the token is right after it. This gives model information about the kind of sentence and how close the token is to the quotation’s essential part. For these features, I needed labels on whether the token is a cue or not. I used gold labels while training the model for feature generation, but I used predictions from the cue classifier for the inference. I showed this dependency in Figure 2.4 with a dashed line.

I used the trained pipelines by SpaCy for the remaining features. SpaCy provides two types of tags; one is more detailed, and the other returns only tags from the list of Universal Part-of-Speech tags.

The features, such as the token’s dependency depth and dependency relation, describe the position of a token in a sentence’s dependency parsing tree. Dependency depth is the number of ancestors of a token. A dependency relation is a tag that describes the relation between a token and the root of a sentence.

The feature “token’s IOB” means a token’s Inside-Outside-Beginning tag produced by the NER pipeline. It is a default NER pipeline implemented in the SpaCy library. The SpaCy’s architecture for the NER pipeline is based on Transition-Based parsing⁸.

Two binary features describe relation with the cue if there is one in the sentence. One feature only carries information if the token is a child of a cue and the other one is more detailed and only True for tokens that are leftmost children of a cue.

Additionally, I used the feature if the token is a quotation mark. I considered it important because quotation marks are essential hints for direct quotations.

L1 and L2 regularization techniques are used together with the CRFs model. I tried Randomized Search Cross Validation for hyperparameters estimation, but default parameters of `c1=0.1` and `c2=0.1` showed the best performance.

⁷<https://universaldependencies.org/u/pos/>

⁸<https://spacy.io/api/architectures#parser>

Source Classifier

The CRF model was used for the source classifier. I used the same features as for the content classifier and added four more features.

1. the label predicted by the content classifier;
2. the token's NER entity type;
3. the distance from the cue, if a token is dependent of one;
4. whether the token is a rightmost child of a cue

It should be noted that for training, I used gold labels of content, not the prediction but the content classifier. I use content classifier predictions for the inference.

Content Resolver

In order to connect identified content span to a cue, I built a content resolver. A resolver is a binary classifier on pairs of cues and content spans, and it returns 1 if a pair does belong to one attribution relation and 0 otherwise. I chose logistic regression as a classification model.

There are the following features for a pair of a content span and a cue:

1. the word-based distance between the content span and the cue;
2. flag if the content span and the cue are in the same sentence;
3. flag if the content span is a descendant of the cue.

If the cue consists of more than one word, I use the token with the shortest path to the root of the sentence or the root itself.

Depending on their relative positions, I calculate the distance from the content span to the cue. If the cue is before the content in the sentence, the distance is the difference between the indices of the first content span's token and the cue. A similar scheme is for the case when the cue is after the content span, but then the final token of the content span is used and absolute value is taken.

For the third feature, I check if some token from the list of the cue's syntactic descendants belongs to the content span; and if it does, then I set the flag as `True` and `False` otherwise.

Source Resolver

The source resolver works similarly to the content resolver. It is a binary logistic regression classifier on the pairs of a source span and a cue.

The following features are utilized:

1. the word-based distance between the source span and the cue;
2. flag if the source span and the cue are in the same sentence;

3. flag if the source span and the cue are in the same parenthetical phrase.

The first two features are defined the same way as for the content classifier, but the third one is different. A parenthetical phrase is a group of words that provides additional information about the sentence but is not essential to its meaning. I extracted these phrases as groups of tokens inside two commas.

I gathered information on all described components in the Table 2.1. It summarizes Sections 2.5 and 2.6.

Component	Model	Number of features	Implementation
Baseline	rule-based	-	based on Textacy
English cue classifier	finetuned NER	-	SpaCy
Czech cue classifier	reporting verbs list	-	original
Russian cue classifier	reporting verbs list	-	original
Content classifier (all features)	CRFs	16	CRFSuite
Content classifier (without text features)	CRFs	12	CRFSuite
Source classifier (all features)	CRFs	20	CRFSuite
Source classifier (without text features)	CRFs	16	CRFSuite
Content resolver	LogReg	3	scikit-learn
Source resolver	LogReg	3	scikit-learn

Table 2.1: Summary of all developed components

In order to see how well the model can work on data in other languages, I test it on the manually annotated corpora of quotations in Czech and Russian. I run experiments with two models: one is trained on all features and another is trained without text features. This approach is considered viable since sentence structures containing quotations exhibit similarities across languages; thus, using features without explicit text may still provide an adequate setup. This is particularly relevant given the scarcity of annotated corpora for quotation extraction tasks in languages other than English.

2.7 Features Analysis

I calculated the information gain to better understand the features' impact on the target feature. I did this analysis only for content and source classifiers because these are the main components together with the cue classifier. The cue classifier is trained differently, so I did not consider it. I excluded text features from the analysis and left only binary and categorical features.

Information gain (IG) is a metric used in machine learning techniques to determine the features that bring more information. It is based on the concept of entropy, which is a measure of the impurity or disorder in a dataset. Information gain is also known as mutual information. The calculation of information gain is done following the formula:

$$IG(T, a) = H(T) - H(T|a), \quad (2.14)$$

where

- T is a target feature,
- a is an estimated feature,
- $H(T)$ is entropy of the target feature,
- $H(T|a)$ is conditional entropy of T given the feature a .

The main idea is to identify the features that provide the most information and contribute the most to the model’s predictive power.

Content Classifier

Feature	Information Gain Value
Flag if the token’s sentence contains a cue	0.237
Flag if the token is inside quotation marks	0.080
Detailed POS tag	0.039
Dependency relation	0.033
Flag if the token follows a cue	0.028
Dependency depth	0.024
POS tag	0.019
Flag if the token is a quotation mark	0.010
Token’s index in a sentence	0.006
IOB tag	0.004
Flag if the token is a child of a cue	0.003
Flag if the token is the leftmost child of a cue	0.002

Table 2.2: Information gain values for the content classifier features

The highest dependency is observed for the feature that describes if the token’s sentence contains a cue (see Table 2.2). The next informative feature tells whether the token is inside quotation marks, which is unsurprising. The features related to the token’s position in the dependency tree have the weakest predictive capacities.

Source Classifier

Results of information gain analysis are provided in Table 2.3.

Information gain scores for source classifier features are given in Table 2.3. The most influential feature is the same as for the content classifier: it is a binary feature that expresses if the token’s sentence contains a cue. The feature of the second importance is the distance from the cue if a token is dependent on one. It can be explained by the word order in English, where a verb is usually right after the subject. Regarding features with low influence on the target feature, the information that the token is a quotation mark or that it is the rightmost child of a cue is among them.

Feature	Information Gain Value
Flag if the token's sentence contains a cue	0.083
Distance from the cue	0.076
Detailed POS tag	0.036
Dependency relation	0.034
POS tag	0.031
Entity type	0.021
Label from the content classifier	0.017
Flag if the token is a child of a cue	0.011
IOB tag	0.011
Flag if the token is the leftmost child of a cue	0.009
Dependency depth	0.009
Flag if the token is inside quotation marks	0.006
Token's index in a sentence	0.006
Flag if the token follows a cue	0.003
Flag if the token is a quotation mark	0.001
Flag if the token is the rightmost child of a cue	0.001

Table 2.3: Information gain values for the source classifier features

3. Results & Discussion

In this chapter, I present the results of three models: a baseline rule-based model and a machine learning system trained on two different sets of features. I tested the models on three languages. I used the PARC dataset for English and the manually annotated test sets for Czech and Russian. Additionally, I evaluated the machine learning-based system on SiR dataset.

A discussion section concludes the chapter, where I give an overview of issues I faced and how they could be solved in future work.

3.1 Baseline Model

I used a rule-based model as a baseline. I describe how it works in Section 2.5. I present the results for each language separately, comparing them with the previously described.

3.1.1 English

The baseline model shows high precision (94-95%) but low recall (13-33%) for all three quotation elements on English data. It means that the model fails to capture a significant portion of the relevant instances. The F1-score, which shows the balance between precision and recall, is relatively low (22-49%), indicating room for improvement in the model’s overall performance. The token-level metrics are presented in Table 3.1.

		En
source	P	95%
	R	13%
	F1	22%
cue	P	94%
	R	23%
	F1	37%
content	P	94%
	R	33%
	F1	49%

Table 3.1: Baseline results on English test set: token level

The measurements on the sentence level (see Table 3.2) answer the question of how well the model detects whether there is a quotation in the sentence. By quotation here, I mean the quoted text, i.e, the content span. The baseline model performs better when evaluated at the sentence level rather than the token level; F1-score equals 67%.

	En
P	79%
R	69%
F1	67%

Table 3.2: Baseline results on English test dataset: sentence level

3.1.2 Czech

As can be seen in Table 3.3, the Czech baseline model shows exceptionally high precision for all three elements of the quote but low recall. So it means that all predicted labels are correct, but only a part of the elements was identified. The F1-score for source and cue is almost the same as for English, but F1-score for content is higher for Czech.

		En	Cz
source	P	95%	100%
	R	13%	12%
	F1	22%	21%
cue	P	94%	100%
	R	23%	23%
	F1	37%	38%
content	P	94%	97%
	R	33%	40%
	F1	49%	56%

Table 3.3: Baseline results on Czech and English datasets: token level

On the sentence level, the results of the baseline model on Czech are similar to the results on English. The sentence-level metrics are better than token-based ones (see Table 3.6). The F1-score on Czech is 65%.

	En	Cz
P	79%	76%
R	69%	68%
F1	67%	65%

Table 3.4: Baseline results on English and Czech: sentence level

3.1.3 Russian

The baseline model shows the best performance on Russian dataset, with the highest F1-scores for every component. However, it still exhibits the overall

		En	Cz	Ru
source	P	95%	100%	90%
	R	13%	12%	18%
	F1	22%	21%	30%
cue	P	94%	100%	95%
	R	23%	23%	31%
	F1	37%	38%	47%
content	P	94%	97%	91%
	R	33%	40%	55%
	F1	49%	56%	69%

Table 3.5: Baseline results on Russian, Czech, and English sets: word level

pattern of high precision and low recall (see Table 3.5). The model identifies the content span better than source and cue, which is also true for English and Czech.

Table 3.6 lists the baseline model performance in detecting sentences with quotations across different datasets. F1-score for Russian is the best among the considered languages.

	En	Cz	Ru
P	79%	76%	77%
R	69%	68%	73%
F1	67%	65%	72%

Table 3.6: Baseline results on English, Czech and Russian: sentence level

The rule-based baseline model performs well when evaluated per sentence, but the performance on individual tokens is average and even low. The model is the most confident with the detection of content spans.

3.2 Machine Learning-Based System

The machine learning-based system consists of five models: cue classifier, content classifier, source classifier, content resolver, and source resolver. I trained two models, a content classifier and a source classifier, with different sets of features. In total, there are two configurations named by their usage of features: with text features and without text features.

I report measurements for each component separately and present the final results of detecting and attributing quotations. The performance of each component is measured on the English dataset for comparison with the results of Newell et al. [2018a]. For the same reason, I considered a predicted span correct if it precisely corresponds to the annotations found in the corpus, the way they do it in their work.

Cue Classifier

For English, I trained a SpaCy’s NER as a cue classifier. I used a list-based classifier for Czech and Russian, due to the lack of training data. The details are described in Section 2.6. The information in Table 3.7 provides metrics of the classifier performance.

	P	R	F1
Cue classifier	81%	85%	78%

Table 3.7: Performance of the cue classifier on English dataset

Content Classifier

The content classifier is a CRF model trained either on the full set of features or on the set without text features (see details in Section 2.6). Both versions of classifiers perform similarly on the English test dataset, as illustrated in Table 3.8. The classifier’s performance on the complete set of features is slightly better; F1-score is 69% as opposed to 65% by the classifier trained without text features.

	P	R	F1
Content classifier with text features	73%	65%	69%
Content classifier without text features	72%	60%	65%

Table 3.8: Performance of the content classifier on English dataset

Source Classifier

The source classifier is also a CRF model; the training features are described in Section 2.6. The source classifier works better than the content classifier. However, it has a similar pattern that the model trained without text features shows worse results (see Table 3.9).

	P	R	F1
Source classifier with text features	78%	78%	78%
Source classifier without text features	74%	74%	74%

Table 3.9: Performance of the source classifier on English dataset

Content and Source Resolvers

Resolvers are logistic regression classifiers that connect a content span and a source span through a cue. Detailed description can be found in sections 2.6 and 2.6. Both the content resolver and the source resolver exhibit strong performance across all three metrics, with precision, recall, and F1-scores ranging between 96% and 98%. The source resolver works slightly better than the content resolver, as presented in Table 3.10.

	P	R	F1
Content resolver	96%	97%	96%
Source resolver	98%	97%	97%

Table 3.10: Performance of resolvers on English dataset

End-to-End Evaluation

I calculated precision, recall, and F1 metrics based on exact matching of full attributed quotations with the annotated data to compare with the performance of the model from the paper by Newell et al. [2018a]. These measures are shown in Table 3.11. The overall performance using this strict evaluation method could be stronger for both Czech and Russian. The values for all metrics are lower than 5%, so I did not include it in the table.

	P	R	F1
Baseline	14%	4%	6%
All features	41%	48%	38%
Without text features	42%	46%	38%
Newell et al. [2018a] results	62%	52%	57%

Table 3.11: Overall performance on English dataset

One token may be associated with more than one label because classifiers for each quotation element work independently, so an isolated evaluation of each model is not enough. I calculated precision, recall, and F1 for each label on the final results of the whole system. Table 3.12 presents these data for all three considered systems.

		Baseline			All features			Without textual features		
		En	Cz	Ru	En	Cz	Ru	En	Cz	Ru
source	P	95%	100%	90%	80%	40%	40%	77%	17%	28%
	R	13%	12%	18%	77%	68%	65%	71%	96%	84%
	F1	22%	21%	30%	78%	50%	50%	74%	29%	42%
cue	P	94%	100%	95%	84%	92%	81%	84%	92%	81%
	R	23%	23%	31%	72%	84%	71%	72%	84%	71%
	F1	37%	38%	47%	78%	88%	75%	78%	88%	75%
content	P	94%	97%	91%	87%	62%	68%	87%	48%	74%
	R	33%	40%	55%	78%	99%	94%	78%	94%	74%
	F1	49%	56%	69%	82%	76%	79%	82%	64%	74%

Table 3.12: Results of three models per language per quotation element

To begin with, I examine the outcomes for each quotation element in a model that utilizes the complete set of features, contrasting it with the baseline model. The machine learning-based system shows a better F1 score for all quotation components, which means it is more balanced than the baseline model. The recall is significantly higher for every experiment. The precision of the model with text features is lower for every component but still relatively high, except for the results on source detection in Czech and Russian.

The original hypothesis was that the model trained on the English dataset without text features would perform better on other languages than a model trained on the full set of features. It can be seen from Table 3.12 with results of the work of all three models that the machine learning system trained without text features does not outperform the system trained on the full set of features. It provides slightly lower metrics in English, but the difference in the other two languages is more significant.

The system without text features is less precise in extracting source tokens in Czech ($P = 17\%$), but the recall is relatively high – 96%. A similar trend exists for Russian: the source recall grows compared to the system with text features, but the precision drops. The opposite situation for extracting content spans in Russian data: recall decreased, but precision increased compared to the system trained on the complete set of features.

Models performances on sentence level quotation detection across different datasets are listed in Table 3.13. Both configurations of machine learning-based systems are better at detecting sentences with quotations for all languages than the baseline rule-based model.

	Baseline			All features			Without text features		
	En	Cz	Ru	En	Cz	Ru	En	Cz	Ru
P	79%	76%	77%	88%	95%	94%	88%	86%	89%
R	69%	68%	73%	86%	94%	94%	85%	80%	88%
F1	67%	65%	72%	87%	95%	94%	86%	79%	88%

Table 3.13: Results of three models per language per sentence

ML-based systems on the English dataset are similar, but the system that uses text features is more potent on Czech and Russian data. The machine learning-based system shows good balance between precision and recall on all languages irrelevant to used features set.

3.3 Evaluation on the SiR Dataset

In an extra experiment, I assessed the machine learning-based systems using the *triple_manual* portion of the SiR dataset (refer to Table 3.14). This dataset provides annotation only for source and cue, so I could access performance only for these two elements.

I do not present an evaluation of the baseline model because there are only three documents with quotation marks, and two of them have an odd number of

them. This is not enough for evaluation.

		All features	Without text features
source	P	20%	15%
	R	42%	56%
	F1	27%	24%
cue	P	78%	78%
	R	48%	48%
	F1	59%	59%

Table 3.14: Performance on SiR dataset

The general trend of the results is similar to the observations on the Czech dataset that I have annotated. Prediction of the source labels has low precision and average recall. The system trained without text features has greater recall but lower precision than the system trained on the full set of features. The scores for cue are the same for both systems because it is the same component based on the list of reporting verbs. The precision is quite good, but the recall is moderate.

3.4 Trained Models Analysis

In order to understand possible reasons for the system’s weak performance, I analyze the parameters of the trained model, such as transition feature coefficients and state feature coefficients. CRFs is a graph based model and it learns coefficients for transition from one state to another (transition coefficients) and weight for particular values of states (state coefficients).

This analysis is done for the content classifier and system classifier trained with text features (see Section 2.6). I focus on these components because they are the most important ones and use CRFs. The analysis is done with tools provided by the Sklearn CRFSuite package. It provides an API to observe transition feature coefficients and state feature coefficients.

3.4.1 Content Classifier Analysis

Let us start with transition feature analysis first. There are three labels (I, O, B), so there are nine possible transitions from one state to another.

As it can be seen in Table 3.15, the most likely transitions are to an inside label from another inside label or from the beginning label. The most unlikely transitions and the ones that should not be possible are from the outside label to the inside label and to the beginning label from another beginning label or inside label. Interestingly, the transition from the outside label to the beginning label is penalized.

In Table 3.16, there are top-5 state feature coefficients. The model strongly connects the beginning of a content span with a preposition *to*. Another observation is that if the detailed POS tag (`spacy_tag`) is equal to opening quotation

Transition	Coefficient
I → I	1.364
B → I	1.356
O → O	0.842
O → B	-4.267
B → O	-6.701
I → O	-8.003
O → I	-12.245
B → B	-12.871
I → B	-15.101

Table 3.15: Transitions coefficients for the content classifier with text features

Coefficient	Label	Feature:value
6.808	B	<code>spacy_tag:T0</code>
5.208	B	<code>sentence_has_verb_cue</code>
5.164	O	<code>spacy_tag:‘‘</code>
4.450	B	<code>prev_5_text:"live fish transporter,</code>
4.450	B	<code>next_5_text:a truck akin to an</code>

Table 3.16: Top-5 positive state features’ coefficients for the content classifier with text features

marks, then it is likely that this token is outside of the content span. Quotation marks are labeled as content in the training dataset, which makes this positive coefficient contradicting.

Coefficient	Label	Feature:value
-6.901	I	<code>follows_verb_cue</code>
-6.199	B	<code>in_quote</code>
-4.295	B	<code>spacy_iob:I</code>
-3.958	B	<code>dependency_relation:acomp</code>
-2.893	O	<code>prev_5_text:Association, a trade group</code>

Table 3.17: Top-5 negative state features’ coefficients for content classifier with text features

I observed negative state features’ coefficients for further insights (see Section 3.17). If a token follows a cue, then the assignment of an inside tag is penalized with the largest negative coefficient. If a token is inside quotation marks, then it is unlikely that the beginning tag is assigned. This is an expected behavior because a prediction for tokens inside quotation marks should be an inside tag. This observation makes the insight from the positive state features’ coefficient related to the opening quotation mark more confusing.

3.4.2 Source Classifier Analysis

As for the content classifier, I analyzed the module trained on the full set of features.

Same as for the content classifier, there are nine possible transitions from one state to another. Transitions coefficients are listed in Table 3.18. Source spans are usually shorter, so this may be the reason why the most probable transition is from the beginning tag to the inside tag, unlike transition coefficients for the content classifier. The transition with the biggest penalty is from the inside tag to the beginning tag.

Transition	Coefficient
B → I	3.922
O → O	1.670
I → I	0.752
O → B	-0.109
B → O	-6.962
O → I	-7.495
B → B	-9.874
I → O	-10.976
I → B	-14.208

Table 3.18: Transitions coefficients for source classifier with text features

Table 3.19 presents top-5 positive coefficients for source classifier features. Reasonably the coefficient for a label ‘outside’ is high when the token was predicted with the beginning label by the content classifier.

Coefficient	Label	Feature:value
7.104	B	sentence_has_verb_cue
6.690	O	content_classifier_label:B-content
4.495	I	prev_5_text:of Red Bank, N.J
4.495	I	next_5_text:-- consented to a fine
4.414	B	prev_5_text:after the government released a

Table 3.19: Top-5 positive state features’ coefficients for source classifier with text features

Table 3.20 gives an overview of the top-5 negative coefficients for the source classifier. The feature with the greatest absolute value of the negative coefficient is a named entity IOB tag. This feature indicates that when a token is inside an entity, it is less likely to be the beginning of a quotation source. The negative coefficient suggests that the model considers it unfavorable for predicting the start of the source.

In conclusion, the analysis reveals that content features are informative for models, enabling them to possess predictive power across different languages.

However, CRFs also give big priority to the text features and some of them have high absolute coefficients. This may suggest that some overfitting is happening. The model trained with both text and content features performs better than the one trained only on content features, likely because including text features intensifies the predictive ability of content features.

Coefficient	Label	Feature:value
-6.523	B	spacy_ent_iob:I
-5.136	I	follows_verb_cue
-4.045	B	dependency_relation:cc
-3.987	O	prev_5_text:(D., Calif.
-3.984	B	spacy_ent_type:DATE

Table 3.20: Top-5 negative state features’ coefficients for source classifier with text features

3.5 Discussion

In this section, I will analyze the results and discuss the challenges I encountered during my work. Moreover, I will outline the possible improvements and directions for future work.

I worked with two systems: a rule-based baseline model and a machine learning-based system. They both were trained on the English dataset, but I tested them on English, Czech, and Russian datasets.

As results show, the machine learning-based system works better than the baseline for detecting quotations’ components in all three languages. The best system is the one where content and source classifiers are trained on both text and not text features. My hypothesis that the classifiers trained without text features would perform better on Czech and Russian data is not confirmed.

It should be noted that the baseline model shows high precision of its predictions but low recall. This rule-based model can capture only direct quotations in quotation marks, which might be a reason for the low recall. The simple system for extraction of indirect quotations would have been able to extract only cues by matching them with the words from the list of reporting verbs. However, it is a too weak anchor for further search of source and content elements without machine learning.

Although the machine learning-based system demonstrates solid results in the extraction of quotation elements, it does not reach good results for the precise detection of quotations. Whereas I followed the model’s description by Newell et al. [2018a], I did not manage to get the same score as they did on the same dataset. They did not provide the code of their work, so reproducing their study was challenging. Without access to their exact implementation, it was difficult to identify potential discrepancies or nuances that could have led to differences in the obtained scores when using the same dataset.

Building a model that would successfully find sentences with direct and indirect quotations, even in other languages, appeared to be a feasible objective. The

problems arise when precision in the extraction of quotations is required. Predicting correct quotation boundaries in indirect quotations is challenging even for the system with the best score. For instance, in Example (1), the model did not include a comma in the content span, so it is a wrong prediction if an exact span only is counted as correct. Using some approximate string matching algorithm for more realistic evaluation could be useful.

- (1) *Big investment banks refused to step up to the plate to support the beleaguered floor traders by buying big blocks of stock, **traders** say.* (wsj_2300)

As a limitation of my work, I can mention the chosen data. The PARC dataset is a valuable source, but it is not ideal for this study. It has too many additional tokens labeled as source and cue, which makes it difficult for the model to focus on important content. The test datasets for Czech and Russian languages can be regarded as relatively limited in size for comprehensive development.

The other potential issue is a possible bias in the detection of cues in Czech and Russian. I used the same reporting verbs lists to form the datasets and also for the cue classifier component. Ideally, the procedure should be independent.

Both as improvement and future development of this work, I see an attempt to train systems for direct and indirect quotations separately. Most of the works in this area are dedicated to direct quotations, so it would have been more interesting to look closer at indirect ones.

Additionally, in the future, I would like to test this model on bigger datasets of languages other than English. It could also be beneficial to experiment with more advanced techniques of multilingual NLP to ensure decent performance on low-resource languages.

Conclusion

Quotations are crucial in media texts, providing insight, context, and support while ensuring journalistic integrity and accuracy. They help represent various viewpoints and enhance transparency and accountability. However, challenges arise when quotes are misattributed or taken out of context, which can mislead readers and undermine credibility. Automatic detection and attribution of quotes can improve accuracy and transparency while minimizing misrepresentation and misinformation risks.

The main objective of this work is to develop a system that would automatically extract direct and indirect quotations and attribute them to their authors in the text. This system should be easily reused on other languages without additional training. I used PARC dataset for the training system on English data and manually annotated Czech and Russian datasets for testing.

As the first stage, I analyzed various types and quotations' structures. The conducted analysis is valuable for a better understanding of how to detect tokens belonging to quotations and devise possible approaches how to handle most of the types.

For the system development, I used ideas from the research for BBC by Newell et al. [2018a]. I built a complex machine learning-based system that consists of five modules. Two main modules use CRFs models. I compared this machine learning-based system with a baseline system created using ideas from the Textacy library.¹

The results indicate that the machine learning-based system outperforms the baseline in detecting quotation components across all three languages. The most effective system is one where content and source classifiers are trained on text and non-text features. However, the hypothesis that classifiers trained without text features would perform better on Czech and Russian data is not confirmed. The baseline model exhibits high precision but low recall, likely due to its focus on direct quotations in quotation marks. Although the machine learning-based system performs decently in detecting sentences with quotations, the precise extraction of tokens belonging to quotations can still be improved.

As one of the outcomes of my work, I developed a tool for quotation extraction and attribution of quotations from English, Russian and Czech sentences. It is available in the GitHub repository associated with this thesis, along with the Czech and Russian datasets, which are annotated for this research and can serve as a basis for expanding the annotated data.²

As a further development of this work, it can be insightful to train systems separately for direct and indirect quotations. Since most research in this area focuses on direct quotations, examining indirect quotations more closely would be interesting. Future work could also involve testing the model on larger datasets of languages other than English and experimenting with advanced multilingual NLP techniques to ensure robust performance on low-resource languages. Other directions for exploration include testing on different datasets, employing deep learning models, utilizing multilingual embeddings for better model versatility,

¹<https://textacy.readthedocs.io/en/latest/index.html>

²<https://github.com/pixelmagenta/adaq>

and addressing coreference resolution to tackle conflicts at the attribution stage.

Bibliography

- Mariana S. C. Almeida, Miguel B. Almeida, and André F. T. Martins. 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*, pages 39–48, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1005. URL <https://aclanthology.org/E14-1005>.
- Carolina Cuesta-Lazaro, Animesh Prasad, and Trevor Wood. What does the sea say to the shore? a BERT based DST style approach for speaker to dialogue attribution in novels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5820–5829, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.400. URL <https://aclanthology.org/2022.acl-long.400>.
- Kira Droganova, O. Lyashevskaya, and Daniel Zeman. Data conversion and consistency of monolingual corpora: Russian ud treebanks. 2018.
- David K Elson and Kathleen R McKeown. Automatic Attribution of Quoted Speech in Literary Narrative. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, page 7, Atlanta, Georgia, 2010. AAAI Press. doi: 10.5555/2898607.2898769.
- The Guardian. Talking sense: using machine learning to understand quotes. *The Guardian*, November 2021. URL <https://www.theguardian.com/info/2021/nov/25/talking-sense-using-machine-learning-to-understand-quotes>.
- Barbora Hladka, Jiří Mírovský, Matyáš Kopp, and Václav Moravec. Annotating attribution in Czech news server articles. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1817–1823, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.193>.
- Ralf Krestel, Sabine Bergler, and René Witte. Minding the source: Automatic tagging of reported speech in newspaper articles. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/718_paper.pdf.
- John Lafferty, Andrew McCallum, and Fernando C N Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001. URL <https://dl.acm.org/doi/10.5555/645530.655813>.
- George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

- Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. A two-stage sieve approach for quote attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 460–470, Valencia, Spain, April 2017. Association for Computational Linguistics. URL <https://aclanthology.org/E17-1044>.
- Chris Newell, Tim Cowlshaw, and David Man. Quote extraction and analysis for news. In *Proceedings of KDD Workshop on Data Science, Journalism and Media (DSJM)*, page 6, New York, NY, USA, 2018a. ACM.
- Edward Newell, Drew Margolin, and Derek Ruths. An attribution relations corpus for political news. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018b. European Language Resources Association (ELRA). URL <https://aclanthology.org/L18-1524>.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL <https://aclanthology.org/L16-1262>.
- Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007. URL <http://www.chokkan.org/software/crfsuite/>.
- Timothy O’Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. A Sequence Labelling Approach to Quote Attribution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 790–799, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL <https://aclanthology.org/D12-1072>.
- Silvia Pareti. A database of attribution relations. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3213–3217, Istanbul, Turkey, May 2012a. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/958_Paper.pdf.
- Silvia Pareti. A Database of Attribution Relations. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3213–3217, Istanbul, Turkey, May 2012b. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/958_Paper.pdf.
- Silvia Pareti. *Attribution: A Computational Approach*. PhD Thesis, University of Edinburgh, Edinburgh, 2015.
- Silvia Pareti. PARC 3.0: A Corpus of Attribution Relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*

- (*LREC'16*), pages 3914–3920, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL <https://aclanthology.org/L16-1619>.
- Silvia Pareti, Tim O’Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 989–999, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <https://aclanthology.org/D13-1101>.
- William Paulo Ducca Fernandes. Quotation Extraction for Portuguese. Rio de Janeiro, Brazil, April 2012. PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO. doi: 10.17771/PUCRio.acad.28807. URL http://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=28807@2.
- Dario Pavllo, Tiziano Piccardi, and Robert West. Quootstrap: Scalable unsupervised extraction of quotation-speaker pairs from large news corpora via bootstrapping. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1), Jun. 2018. doi: 10.1609/icwsm.v12i1.15006. URL <https://ojs.aaai.org/index.php/ICWSM/article/view/15006>.
- Bruno Pouliquen, Ralf Steinberger, and Clive Best. *Automatic Detection of Quotations in Multilingual News*. November 2007. URL <https://publications.jrc.ec.europa.eu/repository/handle/JRC37835>.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/754_paper.pdf.
- Yohanes Sigit Purnomo W.P., Yogan Jaya Kumar, and Nur Zareen Zulkarnain. Understanding quotation extraction and attribution: towards automatic extraction of public figure’s statements for journalism in Indonesia. *Global Knowledge, Memory and Communication*, 70(6/7):655–671, July 2021. ISSN 2514-9342, 2514-9342. doi: 10.1108/GKMC-07-2020-0098. URL <https://www.emerald.com/insight/content/doi/10.1108/GKMC-07-2020-0098/full/html>.
- Andrew Salway, Paul Meurer, Knut Hofland, and Øystein Reigem. Quote extraction and attribution from Norwegian newspapers. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 293–297, Gothenburg, Sweden, May 2017. Association for Computational Linguistics. URL <https://aclanthology.org/W17-0241>.
- Luis Sarmiento and Sergio Nunes. Automatic Extraction of Quotes and Topics from News Feeds. page 12, 2009. URL <https://hdl.handle.net/10216/7080>.
- Christian Stang and Anja Steinhauer. *Handbuch Zeichensetzung. Der praktische Ratgeber zu Komma, Punkt und anderen Satzzeichen*. Duden-Verlag, Berlin, 2 edition, 2014.

- Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2020. URL <https://www.aclweb.org/anthology/K18-2020>.
- Dage Särg, Karmen Kink, and Karl-Oskar Masing. Quote extraction from Estonian media: Analysis and tools. *Eesti Rakenduslingvistika Ühingu aastaraamat Estonian Papers in Applied Linguistics*, 17:249–265, April 2021. ISSN 17362563, 22280677. doi: 10.5128/ERYa17.14. URL <http://arhiiv.rakenduslingvistika.ee/ajakirjad/index.php/aastaraamat/article/view/ERYa17.14>.
- Timoté Vaucher, Andreas Spitz, Michele Catasta, and Robert West. Quotebank: A Corpus of Quotations from a Decade of News. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 328–336, Virtual Event Israel, March 2021. ACM. ISBN 978-1-4503-8297-7. doi: 10.1145/3437963.3441760. URL <https://dl.acm.org/doi/10.1145/3437963.3441760>.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Hovy Eduard, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. OntoNotes Release 5.0, 2022. URL <https://doi.org/10.5683/SP2/KPKFPI>.
- Yuanchi Zhang and Yang Liu. DirectQuote: A Dataset for Direct Quotation Extraction and Attribution in News Articles. Technical Report arXiv:2110.07827, arXiv, October 2021. URL <http://arxiv.org/abs/2110.07827>.

List of Figures

1.1	HMM, linear-chain CRFs, general CRFs	16
2.1	Length of content spans in three datasets	22
2.2	Example of markup in PARC	23
2.3	Diagram of data relations	27
2.4	Machine learning-based system	28

List of Tables

1.1	Datasets statistics	14
2.1	Summary of all developed components	32
2.2	Information gain values for the content classifier features	33
2.3	Information gain values for the source classifier features	34
3.1	Baseline results on English test set: token level	35
3.2	Baseline results on English test dataset: sentence level	36
3.3	Baseline results on Czech and English datasets: token level	36
3.4	Baseline results on English and Czech: sentence level	36
3.5	Baseline results on Russian, Czech, and English sets: word level	37
3.6	Baseline results on English, Czech and Russian: sentence level	37
3.7	Performance of the cue classifier on English dataset	38
3.8	Performance of the content classifier on English dataset	38
3.9	Performance of the source classifier on English dataset	38
3.10	Performance of resolvers on English dataset	39
3.11	Overall performance on English dataset	39
3.12	Results of three models per language per quotation element	39
3.13	Results of three models per language per sentence	40
3.14	Performance on SiR dataset	41
3.15	Transitions coefficients for the content classifier with text features	42
3.16	Top-5 positive state features' coefficients for the content classifier with text features	42
3.17	Top-5 negative state features' coefficients for content classifier with text features	42
3.18	Transitions coefficients for source classifier with text features	43
3.19	Top-5 positive state features' coefficients for source classifier with text features	43
3.20	Top-5 negative state features' coefficients for source classifier with text features	44

List of Abbreviations

PARC – Penn Attribution Relations Corpus
PDTB – Penn Discourse Relations Treebank
CRFs – Conditional Random Fields
NER – Named Entity Recognition
AR – Attribution Relation

A. Lists of reporting verbs

In this attachment I added lists of reporting verbs in English, Czech, and Russian. The origin of this lists and their role is explained in 2.5.

A.1 List of English reporting verbs

accord	estimate
accuse	explain
acknowledge	fear
add	hope
admit	insist
agree	maintain
allege	mention
announce	note
argue	observe
ask	order
assert	predict
believe	promise
blame	recall
charge	recommend
cite	reply
claim	report
complain	say
concede	state
conclude	stress
confirm	suggest
contend	tell
criticize	testify
declare	think
decline	urge
deny	warn
describe	worry
disagree	write
disclose	

A.2 List of Czech reporting verbs

apelovat	appeal
avizovat	announce
charakterizovat	characterize
deklarovat	declare
dodat	add
dodávat	add
doplnit	supplement
hodnotit	evaluate
hovořit	talk
informovat	inform
komentovat	comment
konstatovat	note, observe
kvitovat	acknowledge
líčit	depict
namítat	object
namítnout	object
napsat	write
objasnit	clarify
odmítnout	reject
odpovědět	answer
odvětit	reply
okomentovat	comment
oznámit	announce
popsat	describe
potvrdit	confirm
potvrzovat	confirm

poukázat	point out
poznámenat	note
přiznat se	confess
prohlásit	declare
proklamovat	proclaim
prozradit	disclose
reagovat	react
říci	say
říkat	say
sdělit	tell
tvrdit	assert
upozornit	highlight
upozorňovat	highlight
uvést	state
varovat	warn
vyjádřit se	express
vylíčit	portray
vypovědět	tell, give an account
vyslovit se	declare, speak out
vysvětlit	explain
vysvětlovat	explain
vyzpovídat	confess
zareagovat	react, respond
zdůraznit	emphasize
zmínit	mention
zpovídat	confess

A.3 List of Russian reporting verbs

возражать	argue
выражать	express
говорить	speak
декларировать	declare
дополнить	add
замечать	note
заявить	declare
заявлять	claim
излагать	state
изображать	portray
изобразить	depict
информировать	inform
исповедовать	profess
комментировать	comment
объявить	announce
объяснять	explain
описывать	describe
отвечать	reply
отклонять	reject
оценить	evaluate
передать	deliver
писать	write
подтверждать	confirm
подчеркивать	highlight

подчеркнуть	emphasize
предупреждать	warn
призвать	appeal to
признавать	acknowledge
признаваться	confess
признать	admit
провозглашать	proclaim
произносить	pronounce
разъяснять	explain
раскрывать	reveal
рассказывать	tell
реагировать	react
свидетельствовать	testify
сказать	tell
сообщать	report
спорить	argue
указать	specify
указывать	point
упоминать	mention
утверждать	assert
утверждать	approve
уточнять	clarify
характеризовать	characterize

B. Electronic attachments

The setup for the experiments consists of 13 Python files. I created separate Python files for different components of the systems. These are files for loading and preprocessing of the datasets, files for generating models' features, and converting the training results into the needed form. Additionally, there are files for running the training of the models. These files can be run only for a complete recreation of the workflow. The trained models are available as `.pkl` files.

The cue classifier (called 'verb cue classifier') was trained separately using SpaCy's framework, and it is stored in the folder with the same name. It is also a required component for the experiments. In the beginning stages of the development, I used the name 'verb cue classifier' for 'cue classifier' following naming in Newell et al. [2018a], and I left it in the code.

The electronic attachment consists of two folders: `/quotes_extraction_tool` and `/quotes_extraction_experiments`. The second folder contains folder `/data` with annotated Czech and Russian datasets in the form of `.csv` files with annotated tokens and full sentences.

B.1 Requirements

A full list of required packages is in the file `requirements.txt`. Additionally, it is required to have installed SpaCy language packages `en_core_web_sm` and `ru_core_news_sm`. It can be done with the following commands:

```
python -m spacy download en_core_web_sm
python -m spacy download ru_core_news_sm
```

To evaluate the SiR dataset, the `pybrat` library must be modified since it does not support the relation annotations used in SiR. I provide the modification in the folder `pybrat`, and it can be installed with the following command:

```
cd pybrat && python setup.py install
```

B.2 Usage

B.2.1 Quotes Extraction Tool

Quotes Extraction Tool is a program to extract quotations from the text in English, Czech, and Russian.

The folder `/quotes_extraction_tool` contains the model that showed the best performance (ML-based model trained on all features) and all necessary modules.

The script should be used with these input parameters:

```
python quote_extraction.py 'sentences with quotations'
--lang ['en', 'cs', 'ru']
```

B.2.2 Quotes Extraction Experiments

All experiments were conducted in Jupyter notebooks in the following files:

- `baseline_experiments.ipynb` – evaluation of baseline model
- `ML-based_system_experiments.ipynb` – evaluation of ML-based systems
- `sir_evaluation.ipynb` – evaluation of SiR dataset.

B.3 Developer documentation

Here is a list of all scripts that are needed to reproduce the experiments or to use the quote extraction tool.

List of the scripts and their descriptions

- `baseline.py` — runs baseline model
- `bratcorpus.py` — reads SiR dataset and converts to the compatible format
- `content_classifier.py` — predicts content labels
- `content_resolver.py` — predicts if content and cue belong to the same quotation
- `content_resolver_model.pkl` — content resolver model
- `create_spacy_dataset_for_vcc.py` — forms a dataset for cue classifier training
- `df_corpus.py` — loads Czech and Russian datasets
- `parc3corpus.py` — loads and preprocesses PARC dataset
- `qcc_model.pkl` – content classifier model
- `qsc_model.pkl` – source classifier model
- `quote_resolver.py` — connects together results of content and source resolvers
- `source_classifier.py` — predicts source labels
- `source_resolver.py` — predicts if source and cue belong to the same quotation
- `source_resolver_model.pkl` – source resolver model
- `train_qcc.py` — runs content classifier training
- `train_qsc.py` — runs source classifier training
- `verb_cue_classifier.py` — predicts cue labels