Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Tollef Emil Jørgensen

In a Sentimental Mood

Augmenting Entity-level Sentiment Analysis with Coreference Resolution

Master's thesis in Computer Science
Supervisor: Björn Gambäck
June 2020

NTNU
Norwegian University of Science and Technology

Tollef Emil Jørgensen

In a Sentimental Mood

Augmenting Entity-level Sentiment Analysis with Coreference Resolution

Master's thesis in Computer Science
Supervisor: Björn Gambäck
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

As online media become more prevalent than ever, sentiments towards persons, businesses and other entities spread throughout the world at an increasingly rapid rate. In context of Natural Language Processing, Entity-level Sentiment Analysis is the leading approach to categorize the sentiments expressed towards these entities. Due to the lack of available data, however, research within the field has been left in a stale environment. Therefore, in an attempt to augment the task, this Master's Thesis incorporates Coreference Resolution – the detection and resolving of mentions that refer to a unique entity in a text.

Current systems for Coreference Resolution merely perform evaluations on a single, widely used dataset. Consequently, the usability for Coreference Resolution on other tasks and domains is highly limited. To improve the quality of evaluations, a unified format has been defined. Several datasets are converted into the same, unified format, enabling comprehensive evaluations across domains. A developed framework for Coreference Resolution aims to establish the most generalizable model by evaluating the domain transferability of four model architectures: a deterministic, rule-based model, a statistical model and two based on neural networks. The resulting best model is employed to augment data using an entity-centric segmentation algorithm. A separate framework for Entity-level Sentiment Analysis is used to predict sentiments in the augmented data. This framework comprises four isolated machine learning classifiers: two involving the well established Long Short-Term Memory, an Attention-based neural network, and finally an implementation of the novel Transformer architecture.

Results show that by augmenting larger texts with the help of Coreference Resolution and a segmentation algorithm, models can more accurately predict sentiment towards entities. These results may prove helpful for systems concerning text analytics, especially within domains where distinguishing between positive and negative sentiments is essential, such as for news.

Sammendrag

Utbredelsen av nettbaserte medier er allerede omfattende og utviklingen skjer raskt. Denne utviklingen innebærer også at følelsesbetonte oppfatninger, sentimenter, som omhandler personer, bedrifter og andre entiteter, spres i høyt tempo over hele verden. I kontekst av naturlig språkprosessering er Entitetsnivå Sentimentanalyse den foretrukne metoden for å kategorisere hvilke sentimenter som blir uttrykt overfor entiteter. Mangel på data har imidlertidig ført til begrenset forskningsaktivitet på dette feltet. Ved å endre tilnærmingen til Entitetsnivå Sentimentanalyse, vil denne masteroppgaven involvere koreferansebestemmelse – oppgaven å gjenkjenne og koble sammen uttrykk i en tekst som refererer til en unik entitet.

Nåværende systemer for koreferansebestemmelse utfører kun evalueringer på ett enkelt datasett, med konsekvens at bruksområdet blir innskrenket. For å forbedre evalueringskvaliteten, defineres her et enhetlig format. Flere datasett er konvertert til det samme, enhetlige formatet, som muliggjør omfattende evalueringer på tvers av domener. Et rammeverk for koreferansebestemmelse er utviklet, med mål om å etablere en generaliserbar modell ved å teste domeneoverførbarheten til fire modellarkitekturer: en deterministisk, regelbasert modell, en statistisk modell og to modeller basert på kunstige nevrale nettverk. Den mest egnede modellen vil brukes til å omgjøre data ved hjelp av en entitetssentrisk segmenteringsalgoritme. Et separat rammeverk for Entitetsnivå Sentimentanalyse er brukt til å predikere sentimenter i disse omgjorte dataene. Dette rammeverket omfatter fire isolerte maskinlæringssystemer: to basert på det veletablerte *Long Short-Term Memory*, et basert på hukommelsesmekanismer og et siste på den nyere *Transformer*-arkitekturen.

Resultatene viser at ved å omgjøre større tekster ved hjelp av koreferansebestemmelse og en segmenteringsalgoritme, kan modeller mer nøyaktig utføre sentimentprediksjoner rettet mot entiteter. Disse resultatene kan komme til nytte for systemer som omhandler tekstanalyse, særlig innen domener der det er viktig å skille mellom positive og negative sentimenter, som for eksempel i nyheter.

Preface

This Master's Thesis concludes my Master's Degree in Computer Science at NTNU (Norges Teknisk-Naturvitenskapelige Universitet) Trondheim, Spring of 2020, as described by the course code TDT4900[1]. The thesis mainly concerns the topic of Coreference Resolution, studying its application to Entity-level Sentiment Analysis – motivated by a preliminary specialization project on the latter topic. The title, besides its similarity to Sentiment Analysis, is a reference to both a great tune, dedicated to my jazz-loving younger brother, and to my current state of mind, as this thesis marks the final chapter of my student life.

The thesis has primarily been guided by my supervisor Björn Gambäck, as well as through a collaboration with Strise, a text analytics company in Trondheim. Strise has provided access to systems containing real world media and news events, annotated with information to be used for text mining tasks.

I would like to hand out a special thanks to my supervisor for his excellent domain expertise, as well as for providing me with relevant publications and other material throughout the specialization project and Master's Thesis. Furthermore, thanks to a team at Strise, including Alf Jonassen – for assisting in the annotation of data, Stein-Erik Bjørnnes, Patrick Skjennum and Sigurd Berglann, all supporting me with superb knowledge and hands-on experience with several topics needed to complete this thesis. Lastly, thanks to my mother for assisting with the abstract and title, and to Kristine, for keeping me company and encouraging me during the writing.

Tollef Jørgensen
Trondheim, 11th June 2020

[1] https://www.ntnu.edu/studies/courses/TDT4900

Contents

List of Figures

List of Tables

Acronyms

CL-Eval the CorefLite Evaluation Framework. 4, 70, 71, 78

CR Coreference Resolution. 1–5, 23, 25–27, 29, 31–35, 38–43, 47–50, 59, 64, 70, 73–75, 79–83, 89–94, 96, 99–104, 107, 109–115

CRF Conditional Random Field. 15, 80

DS Distant Supervision. 3, 43, 89, 92, 94, 100, 109

Elsa-Val the Entity-level Sentiment Analysis Framework. 70, 90, 92, 93

ESA Entity-level Sentiment Analysis. 1–3, 5, 43, 45, 52, 59, 70, 89, 90, 94, 99, 101, 102, 104, 107, 109, 111

LM Language Model. 20

LSTM Long Short-Term Memory. 18, 68, 80

MRC Machine Reading Comprehension. 40, 42

NER Named Entity Recognition. 10, 29, 80

NLP Natural Language Processing. 1, 7, 15, 20, 29, 31, 32, 38, 46, 93, 99

NN Neural Network. 15, 16, 18–20, 38, 39, 42, 109

NP noun phrase. 37, 114

POS part-of-speech. 9, 10, 115

RNN Recurrent Neural Network. 17–19

SA Sentiment Analysis. 1, 14, 34, 41, 45, 47, 93

1. Introduction

Determining how entities (e.g. persons, businesses, locations) are represented online is of great use for several applications, such as tracking how a company is portrayed in the media and retrieving opinionated information on political campaigns and other events. In order to solve these problems, Sentiment Analysis (SA) – the task of classifying opinionated text – plays a vital role. In its simplest form, SA provides little detail on its predictions, as there is no specified target of the opinion. To improve upon traditional SA, targets – often called named entities – will first have to be recognized, before obtaining relevant text to compute the targets' sentiment polarity. This is referred to as Entity-level Sentiment Analysis (ESA). Current approaches use machine learning systems to detect scopes containing the required text to represent an entity (Li and Lu, 2017, 2019), disregarding the possible benefits from implementing semantic heuristics, such as those generated with the help of Coreference Resolution (CR). CR defines the process of discovering and resolving mentions that refer to the same entity in a document, a technique that may be used to enrich other high-level tasks of Natural Language Processing (NLP). In Example 1, the functionality of a CR system is illustrated. The pronoun "her" refers to "Anna", "he" to "John", and "it" to "bike".

Example 1 $\text{Anna}_{(1)}$ bought $\text{John}_{(2)}$ a new $\text{bike}_{(3)}$. $\text{He}_{(2)}$ told $\text{her}_{(1)}$ $\text{it}_{(3)}$ was great!

While trivial, the example shines light on the versatility of CR. For instance, we can observe its usefulness in context of SA: "it was great" can be parsed as "the bike was great". The former sentence would provide no meaning without resolving the antecedent of "it". Despite the observed importance of CR and the major improvements discovered in recent research, CR models are seldom found to be implemented in state-of-the-art solutions in other NLP tasks. Moosavi (2020) hypothesizes that the lack of robustness in CR systems is the culprit – leaving the models unable to generalize well to out-of-domain data. This may be due to the lack of a standardized format in current CR datasets, as it hinders researchers to include these datasets in their evaluations. Addressing robustness, this Master's Thesis presents a coreference evaluation framework, designed to convert a selection of datasets into a unified format and perform comprehensive evaluations, never before published in literature. The evaluation process involves four different CR models: a deterministic rule-based model, a statistical machine learning model, and two neural network models. These models are all evaluated thoroughly, with the goal to discover the most generalizable CR model. The resulting model is employed to augment larger pieces of text using an adaptable, entity-centric segmentation algorithm, extracting relevant phrases corresponding to each entity. Finally, a selection of four ESA models of varying complexity are evaluated using the generated segments of text, leading to

improved accuracy on the task of predicting entity-level sentiment. As will be revealed, neural networks using pre-trained language models perform indisputably better than the alternatives, both for CR and ESA.

1.1. Background and Motivation

The topic of Entity-level Sentiment Analysis (ESA) was extensively researched in a preliminary specialization project (Jørgensen, 2019). One research question was defined as follows: "Given a set of entities in a document, how can sentiment be connected towards each respective co-referenced entity?", which sparked interest in the field of Coreference Resolution (CR) for continued research. No methods uncovered in the specialization project had discussed the possible benefits of incorporating CR, which was found to be surprising, given the intuitively idealistic relationship between the two fields – both aiming at resolving text connected to entities. In recent times, the research of ESA has been left in a stale environment (Pei et al., 2019), possibly due to the lack of real world data. Leading state-of-the-art methods still rely on small datasets comprising online reviews and Twitter posts, providing little use for applications in other domains – especially those of formal text.

To combat the lack of data, a dataset has been annotated by distant supervision, based on information obtained from a knowledge graph accessible through a collaboration with Strise – a text analytics company in Trondheim. The dataset includes a large variety of online publications, mostly from news sources, across a selection of higher level topics. Additionally, the knowledge graph contains information on recognized entities and their respective aliases (e.g. MS for Microsoft) and relations (e.g. a CEO-relation for Microsoft: Satya Nadella).

1.2. Goals and Research Questions

The overall goal of the Master's Thesis is defined as follows:

Goal *Establish a well-generalized Coreference Resolution model to augment the task of Entity-level Sentiment Analysis*

By evaluating existing Coreference Resolution (CR) models on in- and out-of-domain data, a desired model architecture may be discovered for generalizable CR. Using this model, research and discuss its impact on Entity-level Sentiment Analysis (ESA). Below are a set of research questions related to the process of reaching the goal:

Research question 1 *How well do Coreference Resolution models perform when evaluated on out-of-domain data?*

By using a diverse set of models found in literature, evaluate them on a selection of in- and out-of-domain datasets. Through this process, the most generalizable model may

be discovered. The generalizability has previously been addressed as a great concern in order for a model to apply well across other domains of text (Moosavi, 2020), which results in better applications for other tasks – such as ESA.

Research question 2 *Can current datasets for Entity-level Sentiment Analysis be used as out-of-domain evaluation baselines?*

In the specialization project, a severe lack of data for ESA was discovered. Currently, the datasets by Pontiki et al. (2014) and Dong et al. (2014) are still used for state-of-the-art models (Rietzler et al., 2020), which only regard the domains of online reviews and Twitter messages. To aid research in other domains, new datasets must be created. Without the resources to manually label a sufficiently large in-domain dataset, however, research whether existing datasets can be used as evaluation baselines. Additionally, to test these baselines, create a dataset using knowledge graphs and world knowledge to mimic entity-level sentiment, annotating using Distant Supervision (DS). The ideal result is to be able to evaluate the DS-annotated dataset on existing ESA datasets. If this process yields any positive results, more resources can be put into the creation of automatic, large-scale datasets for ESA – benefiting the field greatly.

Research question 3 *Can augmentation of datasets result in improvements using Entity-level Sentiment Analysis models?*

By augmenting datasets using CR, the amount of available labeled data will increase, while simultaneously contributing to disordering of data – as it will diverge from its original state. Study the results of transforming data for the ESA task and uncover possible hindrances or improvements with this novel technique.

1.3. Research Method

An experimental methodology is used, as several experiments are required to pursue the goal of the Master's Thesis. The experiments are carried out in a similar manner as to those found in related literature, with the addition of datasets that have not yet been evaluated. The approach involves following the traditional evaluation metrics, as well as incorporating a newer metric by Moosavi and Strube (2016) that addresses the issue of generalizability in Coreference Resolution (CR). For experiments on Entity-level Sentiment Analysis (ESA), an in-domain dataset is labeled by Distant Supervision (DS), using a data-rich knowledge graph. This data is tested and evaluated on existing datasets, both to verify the integrity of generated data, as well as the capability of existing datasets as evaluation baselines. Hardware-intensive experiments and evaluations have been run on the NTNU IDUN computing cluster (Själander et al., 2019).

1.4. Contributions

1. A thorough evaluation of Coreference Resolution models on a variety of datasets

2. A defined, *light*, unified format for Coreference Resolution annotation – coined CorefLite
3. An open-sourced tool to convert Coreference Resolution datasets into CorefLite
4. An approach to create Entity-Level Sentiment Analysis datasets using knowledge graphs and distant supervision
5. Experiments on the augmentation of data for for Entity-Level Sentiment Analysis using Coreference Resolution

Openly available systems are summarized below, with URLs to the GitHub repositories where further code-specific information can be found. Raw data accessed from the knowledge graph – specifically event texts – can be given upon request.

CorefLite Converter

The CorefLite converter is built into the the CorefLite Evaluation Framework (CL-Eval) system below. `https://github.com/ph10m/CorefLite`

Coreference Resolution Evaluation Framework (CL-Eval)

`https://github.com/ph10m/ClEval`

Entity-level Sentiment Analysis Framework (Elsa-Val)

`https://github.com/ph10m/ElsaVal`

1.5. Thesis Structure

The thesis contains a total of nine chapters. Below are a list of the respective chapters and their primary purpose.

1. **Introduction**
   Give the reader an introduction to the purpose and goals of the thesis, as well as an overview of contributions.
2. **Background Theory**
   Presents background theory closely related to the topics to be covered throughout the thesis.
3. **Related Work**
   Starting with a customized structured literature review, the basis for selected literature is documented (with additional material in Appendix A). The rest of the chapter is dedicated to presenting related work in the field of Coreference Resolution (CR).

4. **Data**
   This chapter regards the available datasets for CR and Entity-level Sentiment Analysis (ESA). An inspection and analysis of the data results in a selection of relevant datasets to be used further. Additionally, the process of generating a dataset for ESA with Distant Supervision is documented.

5. **Architecture**
   Here, architectures for developed and used systems are presented. Many visualizations are included to give the reader a good overview of the higher level functionality of systems and frameworks used.

6. **Coreference Validation**
   As the first of two experimental chapters, Coreference Validation includes the process of evaluating CR models and validation of datasets converted to the CorefLite format, resulting in a defined well-generalized CR model.

7. **Entity-level Sentiment Analysis**
   Experiments conducted on ESA, with and without augmented data using the previously defined CR model. A baseline for existing data is set up, for accurate evaluation of the generated dataset from the Data chapter. Additionally, a manually labeled dataset is evaluated and augmented.

8. **Evaluation and Discussion**
   Contains evaluation of the research questions and goal, as well as discussions on the topics of CR and ESA.

9. **Conclusion and Future Work**
   The final chapter concludes the work done in this thesis, presents the most worthy contributions in more detail, and ends with suggestions for future work.

2. Background Theory

Natural Language Processing (NLP) is the common term used to describe the interaction between computers and natural languages. The main goal of any NLP system, as stated in Gambäck et al. (1994), is making computers able to interpret any given utterance in a natural language. This chapter starts by conveying the very basics of NLP, progressing towards more specialized topics. Some sections are reused from the specialization project (Jørgensen, 2019), and these will be clearly identified.

2.1. Introductory Topics for Natural Language Processing

Natural languages are unspecific, flexible, and full of redundancies and ambiguities. If natural languages were to be handled directly in computer systems, they would quickly become cumbersome to deal with. In order to simplify the process of analyzing the languages we speak and write, applying techniques to preprocess and represent text by other means can be of great help – some of which are covered here. This section (2.1) has been reused from the specialization project (Jørgensen, 2019), as it still works as a great foundation for basic understanding of NLP.

2.1.1. Text Preprocessing

Stop Word Removal

Words that frequently appear across a set of documents typically contribute negligible discriminative value to the given documents, and are commonly removed. For English, this may be words like “a”, “it”, “the”. Lists of stop words can be found in programming libraries for text processing, such as Natural Language ToolKit (NLTK)[1] (Loper and Bird, 2002) and spaCy[2] (Honnibal and Montani, 2017).

Normalization

Normalization has the purpose of improving predictability and reducing ambiguity - transforming the text before processing it further. Normalization and related topics are covered in detail by Mikheev (2000). Examples of text that should be handled can be seen in Table 2.1. Two common techniques to handle grammatical inflection are stemming and lemmatization. Stemming is the removal of inflectional endings from words, getting rid of any affixes. Lemmatization is in essence stemming with dictionary lookup. However,

[1] https://github.com/nltk/nltk
[2] https://github.com/explosion/spaCy

Inflected words	ask, asking, asked
Capitalization	"I'm on the verge of ..." "I read it on *The Verge*"
Repeating letters	Looooooooong
Punctuation	e.g., U.S.A, really?!
Spacing and grouping	"Hong Kong", "the man"

Table 2.1.: Text to be handled by normalization

Lemmatization	am, are, is $\longrightarrow$ be
Stemming	sensation $\longrightarrow$ sensat owned $\longrightarrow$ own

Table 2.2.: Techniques for handling inflection

lemmatization has the goal of reducing a word to its base or dictionary form (known as its *lemma*). Examples of lemmatization and stemming can be seen in Table 2.2.

2.1.2. Document Representation

A text, in context of NLP, is often referred to as a *document*. The representation of a document considers how textual data is fed into the computer program. Following are a few possible ways to represent documents, using document D as an example throughout the Section: "NLP is exciting, and is one of many fields of AI" (D).

Bag-of-words

Bag-of-words represents textual content as a vector with values corresponding to the total count of each unique word. The total size of the vector will equal the amount of unique words in the vocabulary. An example with document D:

input: `"NLP is exciting, and is one of many fields of AI"`
count: `[NLP: 1, is: 2, exciting: 1, and: 1, one: 1, of: 2, many: 1, fields: 1, AI: 1]`
output: `[1, 2, 1, 1, 1, 2, 1, 1, 1]`

N-Grams

Using n-grams, a document is represented in batches of N-tuples. The N describes the amount of words batched together. Common namings of N-values include unigrams,

bigrams or trigrams (batches of 1, 2 and 3, respectively). This can help identify relations between words such as "Sherlock Holmes" (bigram) and "Natural Language Processing" (trigram) in corpora. The document D has the following representation using a bigram model:

```
[(..., NLP), (NLP, is), (is, exciting), (exciting, and), (and,
is), (is, one), (one, of), (of, many), (many, fields), (fields,
of), (of, AI), (AI, ...)]
```

Furthermore, N-gram models can also be represented as bag-of-words models, in which unigrams are the counted term.

TF-IDF

TF-IDF, conceived by Spärck Jones (1972), is a well established statistic in the field of information retrieval (IR), and is used to establish relative importance of terms in documents in a corpus. TF-IDF is composed of two separate IR techniques, Term-Frequency (TF) and Inverse Document Frequency (IDF). Term-Frequency refers to the number of times a term occurs in a document, and Inverse Document Frequency is a score that adjusts the importance by accounting for how frequent the word is in the corpus. In short, TF-IDF will not attribute much importance to equally common words across the corpus.

Annotations and Tagging

Annotations are used to further define the contents of text by including information like tags, structure and semantics to terms in documents. This is commonly called *tagging*. One popular annotation technique is part-of-speech (POS) tagging, which assigns syntactic functions (grammatical relations) or part of speech to each respective term. The main idea is to be able to differ between abbreviations and ambiguous terms like "can", which can take multiple forms: "can" [verb], "can" [noun], "Can" as in Canada, "Can", a Turkish name, "CAN" as in "CAN bus", a micro-controller communication system for vehicles.

2.1.3. Language Models

A language model, or a *statistical* language model, specifies a probability distribution over sequences of terms, typically on a word-level (Wang and Zhai, 2017). Similar to the N-gram representation in 2.1.2, the first language models were based on the prediction of the next N-gram in a text, developed by Katz (1987). An optimal language model will with confidence predict the next occurring term in a document, based on its previous observations of terms frequently occurring together.

2.1.4. Word Embeddings

Word embeddings are used to represent words as vectors, mapped from a multi-dimensional vector space to a much lower dimension. The contents (or features) of the reduced vector (the *embedding*) of a word can include information about its semantics, context and much more, based on its relation with other words with similar distributions – closely related to the distributional hypothesis (Harris, 1954). The features of a vector may include underlying information of the word, such as:

```
King = ['monarch':  1, 'man':  1, 'woman':  0]
Queen = ['monarch':  1, 'man':  0, 'woman':  1]
```

A commonly used example is the application of mathematical operators on the features of word embeddings:

$$King - Man + Woman = Queen$$

In order to construct the embeddings, a popular approach is *word2vec*, developed and described in detail by Mikolov et al. (2013). More recent embeddings, commonly used in modern systems are GloVe (Pennington et al., 2014), ELMo representations (Peters et al., 2018) and BERT embeddings (Devlin et al., 2019).

2.2. Core Topics

These are topics closely related to the rest of the Master's Thesis. They assume decent understanding of the previously covered sections. Sections 2.2.1 and 2.2.2 are reused from the specialization project (Jørgensen, 2019). The rest are new additions.

2.2.1. Named Entity Recognition

Named Entity Recognition (NER) is the task of recognizing entities in a document. An entity is a product, service, topic, person, organization, issue or event (Liu, 2017). As with annotations (Section 2.1.2), entities are often tagged with its entity type. Using the sentence "Mr. Apple, who worked at Apple, ate an apple", an optimal system will identify the two entities `Apple[PERSON]` and `Apple[ORGANIZATION]`, and a POS tagger (also specified in Section 2.1.2) should identify `apple[NOUN]`. Two important aspects of NER for this project are named entity disambiguation and relation extraction, briefly described below.

Disambiguation

An entity (such as a company or person) may be written about using acronyms or aliases. The U.S. politician Alexandria Ocasio-Cortez is commonly called "AOC" in the media. AOC is also the name of a consumer electronics company, and is thus an important reference to resolve.

Relation Extraction

Relation extraction is the task of extracting relations between entities in a document. In the previously used sentence "Mr. Apple, who worked at Apple, ate an apple", a relation extraction system should identify that `Apple[PERSON]` is an employee of `Apple[ORGANIZATION]` by the relation `works_at` or similar.

2.2.2. Sentiment Analysis

Sentiment analysis and opinion mining are generally used to describe the same topic. Liu (2012) describes it as a multi-faceted problem, to be considered as the computational study of people's opinions, appraisals and emotions toward entities, events and their attributes.

Entity-Level Sentiment Analysis

Entity-level sentiment analysis, first introduced by Moilanen and Pulman (2009), is the task of classifying sentiment with respect to target entities in a document. Its objective, according to Liu (2017), is to discover all opinion quintuples *(e, a, s, h, t)*:

e: target entity

a: aspect of **e**

s: sentiment on aspect **a** of target **e**, consisting of the sentiment in a value range to reflect both orientation and intensity, e.g. $(0, 1, ..., 10)$

h: holder of the opinion

t: time of expression

Aspect Extraction

An aspect is a feature or an attribute of an entity, such as *price* for the mention of whether a product is expensive. Typically, entity targets are explicitly stated (or as a reference), making for relatively easy extraction. Aspects, however, are implicit, as they are inferred from the contents of the document. The sentence "My phone takes terrible photos" is an implicitly negative sentiment on the aspect `camera` of entity `phone`.

Negation in Sentiment Analysis

Negation is the process of reversing a classified sentiment score. Negation can be found in several grammatical forms, such as in affixes ("e.g. *im*possible", "*non*-functional"), content-words (e.g. "not", "never") and function-word (e.g. "eliminated", "reduced") (Choi and Cardie, 2008). Content-word negation and function-word negation may be considered *syntactic* negation, in which a set of words are negated by a word or phrase. For NLP, syntactic negation is of most interest, as the negation modifies the related text

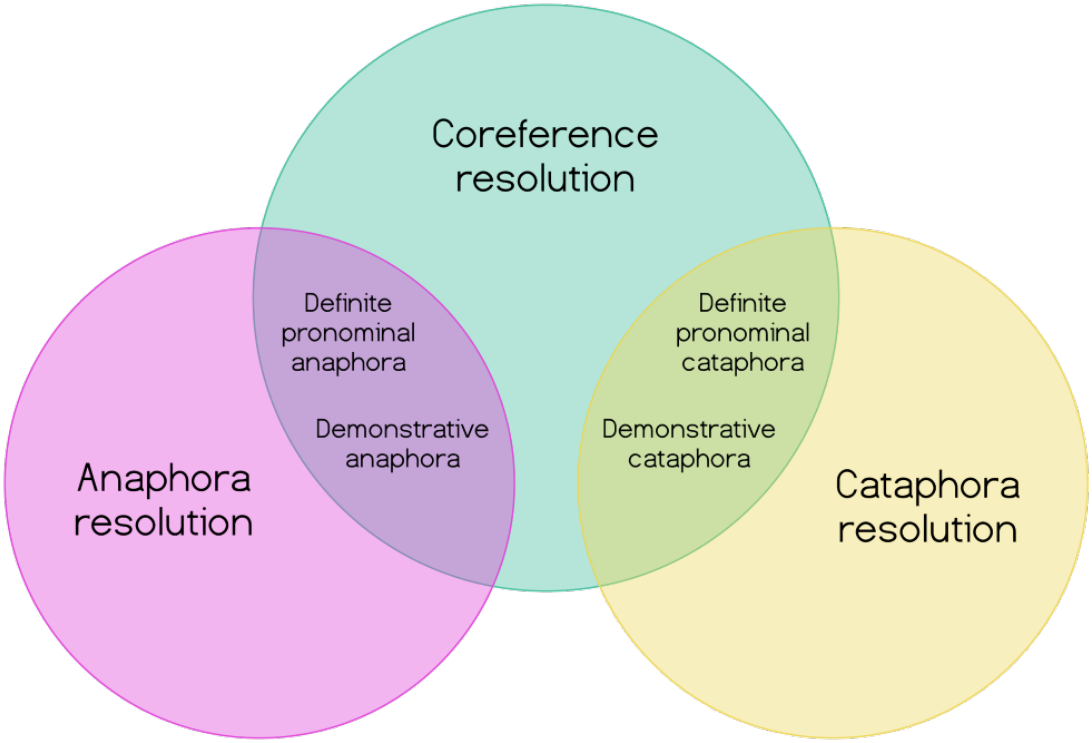


Figure 2.1.: Venn diagram showing overlapping types of reference resolution

entirely, whereas words negated by affix-negation are implicitly negative. Reitan et al. (2015) covered negation extensively in the development of a negation classifier.

2.2.3. Coreference Resolution

Coreference resolution aims to identify which phrases or mentions that refer to the same real-world entity or concept (Rahman and Ng, 2009). "Adam waved to Anna, she waved back! He asked her to walk with him to school". Here, multiple references to "Adam" are present. Referring to an entity often relies on resolving its antecedent – a word or phrase that is the root ancestor of the reference. References can appear in several shapes and forms, such as with demonstratives or presuppositions. How the different types of references relate is illustrated in Figure 2.1.

Anaphora

Anaphora are references that refer back to an entity mentioned earlier in a piece of text or discourse. Anaphoric expressions can be defined as an intralinguistic terminology (Sukthanker et al., 2018), as all references are present in the text itself, thus they do not require world knowledge to resolve.

Cataphora

References to an entity before it is mentioned. "After he was received the phone call, John ran home". More complex occurrences of cataphora requires excessive use of extralinguistic features to resolve.

Split references

Both anaphoric and cataphoric expressions may involve subject pronouns regarding multiple targets (e.g. they, them). An example with anaphora: "Adam and John had finished their chores, so they watched TV". An example with cataphora: "He's in the kitchen making them now, if you want cookies".

Demonstratives

When an entity is not explicitly specified, but referenced through a demonstrative. "He said he liked this phone much better than that[0] one". [0] refers to an implicit entity (a phone).

Definite Pronominals

References using definite pronouns (e.g. his, her, me, you, I) – "Adam was walking up the stairs when he fell". Definite pronominal references target a unique entity. Can occur for both anaphoric and cataphoric references. Early work in reference resolution focused strictly on the task of pronominal resolution, as that of Hobbs (1978); Roberts (1989).

Presuppositions

References happening in context of indefinite pronouns (e.g. someone, somebody, anyone) are used within a document – "Almost all the firemen had to help out". These are references to an unspecific entity or group of entities. Projection of presuppositions as a resolution task was first introduced by Van der Sandt (1992).

2.2.4. World Knowledge and Knowledge Bases

World knowledge regards knowledge that seemingly only humans possess. A commonly used demonstration of the need for world knowledge is the Winograd Schema Challenge (Levesque et al., 2012) – a test containing sentences in which one or more referential ambiguities are present, as built upon by the work of Winograd (1972). Considering Winograd Example 1 and 2, the adjectives *big* and *small* completely modify the reference to "it". The only solution to this is incorporating knowledge of the two objects "trophy" and "suitcase".

Winograd Example 1 *The trophy doesn't fit in the brown suitcase because it's too big.*

Winograd Example 2 *The trophy doesn't fit in the brown suitcase because it's too small.*

A system used to provide world knowledge is often referred to as a "knowledge base" or "knowledge graph".

Sources for Knowledge

A few commonly used knowledge bases are Yago (Suchanek et al., 2007), FrameNet (Baker et al., 1998) and WikiData (Vrandečić and Krötzsch, 2014). These contain extratextual information, such as what an object is a subclass of (e.g. car is a subclass of vehicle), who the spouse for a famous politician is and where a company resides. This data can, if used properly, help resolve references.

2.3. Classification of Natural Language

This section, up until the section on pre-training (p. 20), has been reused from the specialization project. The reused sections are still deemed relevant for understanding the classification of Sentiment Analysis (SA). Three types of classification methodologies will be presented: lexicon-based, supervised learning and pre-training.

2.3.1. Lexicon-Based

Lexicon-based methods do not require any statistical input data, but rely on lexica consisting of words weighted on sentiment orientation and more (Saif et al., 2016). An example of such a lexicon is SentiWordNet (Esuli and Sebastiani, 2006). Although lexicon-based methods require frequent revision by humans to stay up-to-date and relevant, a costly task, they tend to perform well when applied to different domains due to precise connections to semantic composition and linguistic features. A great weakness, however, is that lexicon-based techniques rely on prior sentiment; words have an attributed meaning before they are placed in context of a sentence.

2.3.2. Supervised Learning

Supervised learning methods consist of two phases: training and prediction. During training, a machine learning algorithm requires labeled training data, along with a set of features, in order for it to spot patterns in the input data. The result of this process is a trained classifier, able to create a prediction on new unlabeled data. An illustration of the training and prediction process can be seen in Figure 2.2. Supervised learning has its limitations, one of which is domain dependence. Classifiers trained specifically on data from news may produce unsatisfactory performance applied to data from other domains (Aue and Gamon, 2005). Today, however, we see that classifiers can be generalized across domains with the usage of pre-training and large language models (Radford et al., 2019), covered in the next section, p. 20. Below are a few approaches to supervised learning classification, including Support Vector Machines – used in early models for Sentiment Analysis (SA), Conditional Random Fields – which have been successful in

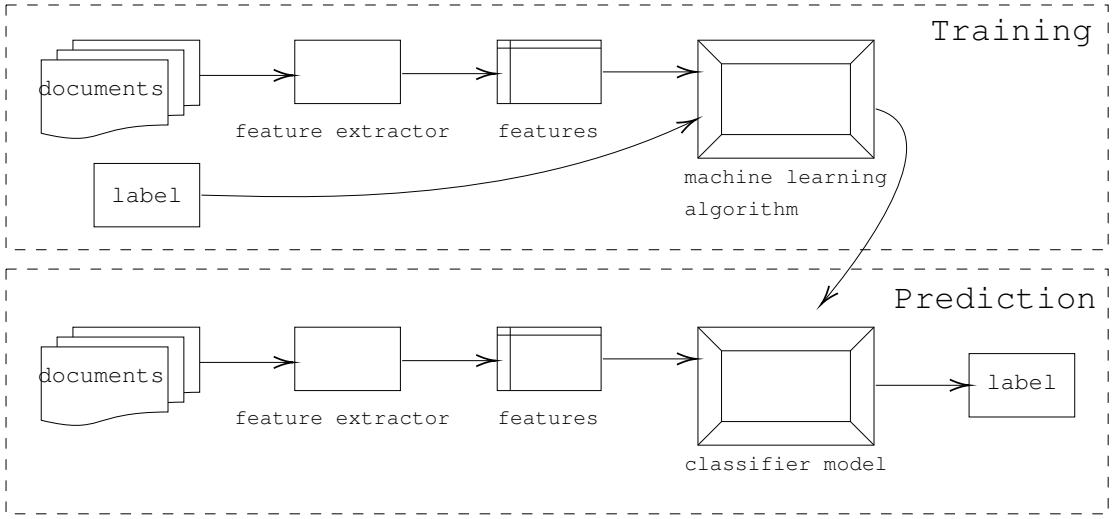


Figure 2.2.: The training and prediction phase of supervised machine learning

aiding advanced models with heuristics, as well as a brief introduction to Artificial Neural Networks and commonly used deep learning architectures.

2.3.2.1. Support Vector Machines

The purpose of a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) is to create a mapping of the data to a higher dimension, such that it is possible to draw a hyperplane called the *support vector* to separate the higher order data points by drawing a *support vector classifier* with a goal of maximizing the margin around the separation, resulting in the best possible split for a classification. Separating data is tricky, as the separation needs to be transformed by a mathematical function. This transformation is computed by kernel functions, introduced by Boser et al. (1992), whose primary functionality is finding support vector classifiers for data as if it were of a higher dimension. Kernel functions accomplish this by calculating point-wise relationships between all data points, such as the polynomial kernel, applying p^d for a point p and dimension d.

2.3.2.2. Conditional Random Fields

Conditional Random Fields (CRFs), presented in Lafferty et al. (2001) are undirected graphs used to build probabilistic models for segmenting and labeling sequence data, largely guided by the fundamental theorem of random fields (Hammersley and Clifford, 1971). CRFs, in the context of Natural Language Processing (NLP), have been used in several high-performing methods, usually as a stochastic heuristic combined with Neural Networks (NNs) to create a final classifier.

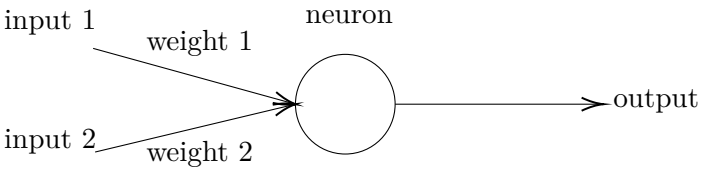


Figure 2.3.: A neuron in a neural network

2.3.2.3. Artificial Neural Networks

The functionality of an Artificial Neural Network, commonly referred to as an NN, is inspired by how our brains work and learn – the biological neural network. In its simplest form, an NN is composed of an input, a binary classifier called the perceptron, first introduced by Rosenblatt (1958), and an output. As seen in Figure 2.3, each of the inputs to the neuron (perceptron) have an assigned weight, where the neuron computes an aggregation of all its inputs and weights.

Activating neurons An activation function defines how the input data is handled in a neuron, before passing it on to the next layer in the network – thus defining how the neuron is *activated*. Activation functions are often categorized as linear or non-linear – depending on how they transform the data. Some commonly used activation functions are the Sigmoid function, hyperbolic tangent (tanh) and the rectified linear unit (ReLU). Only the Sigmoid function will be referenced in this paper. For the interested reader, more can be found in Nwankpa et al. (2018) and Goodfellow et al. (2016). The Sigmoid function has been used extensively since the early days of neural networks, especially those regarding binary classification (such as sentiment polarity values -1 or 1). It may also be called the *logistic* function, due to its definition:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

When handling an input x, the Sigmoid function transforms the input to values between 0.0 and 1.0, ensuring the output of a neuron is in a predictable range.

Altering input weights The objective of an NN is to discover optimal statistical patterns between the inputs and outputs. This is done by altering the input weights throughout the network as the perceptrons are activated. The altered weights are modified with respect to a *loss function*. A loss function determines how the error (difference between desired output and guessed output) should be calculated. Weights are updated based on the current error and a *learning rate* (defining how much the error should influence the updated weights). To create a network, several neurons are set up in layers, referred to as hidden layers. Inputs enter the hidden layer, and the activated neurons pass data along in the network. A simple illustration of an NN is shown in Figure 2.4.

Typically, several hidden layers are used. What has been shown here is commonly referred to a feedforward neural network, as all the outputs from each neuron are passed

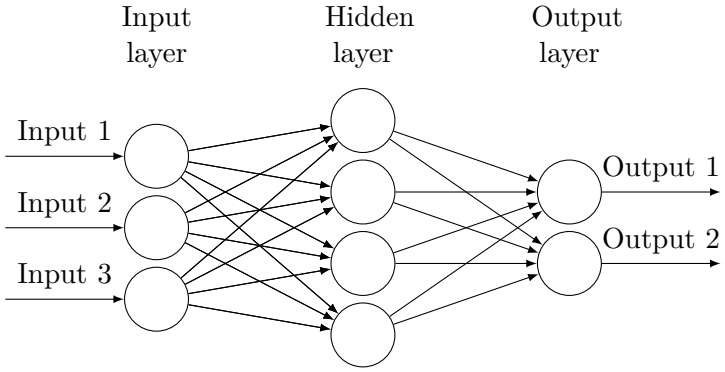


Figure 2.4.: Feedforward neural network

along to the next layer (seen in Figure 2.5). As shown in the next few sections, the outputs from each layer can be set up in more complex structures such as those found in recurrent neural networks.

2.3.2.4. Recurrent Neural Networks

Where the previously covered feedforward neural networks simply pass along the output of neurons, Recurrent Neural Network (RNN)s pass the output into another layer using data from the current state of the hidden layer – this enables them to "remember" earlier learned outcomes, as opposed to starting from scratch at any given point in time. The new hidden state at a given time t for a function f with parameters W, f_W, an old state h_{t-1} and input at a given time, x_t, can be expressed as

$$h_t = f_W(h_{t-1}, x_t)$$

Each step of this procedure is called a *cell*, which handles this computation and calculates an output and loss – depending on its configuration. The outputs can be used to combine the sequence of cell output data for a final classification. RNNs were developed to address long-term dependencies, but proved to be inefficient for dependencies stretching far outside the beginning or end of a sequence. It is not feasible to store numerous previous data points back in time, as this requires enormous amounts of memory. Moreover, the data passed from each cell becomes cluttered, such that there is no way to extract old information from the data.

Due to the architecture of RNNs with sequential cells, two problems arise: exploding gradient and vanishing gradient. The exploding gradient may cause multiplication of numbers to increase exponentially between each sequence of cells. Vanishing gradients are essentially describing the same process, except for small numbers, converging to zero (Hochreiter and Schmidhuber, 1997).

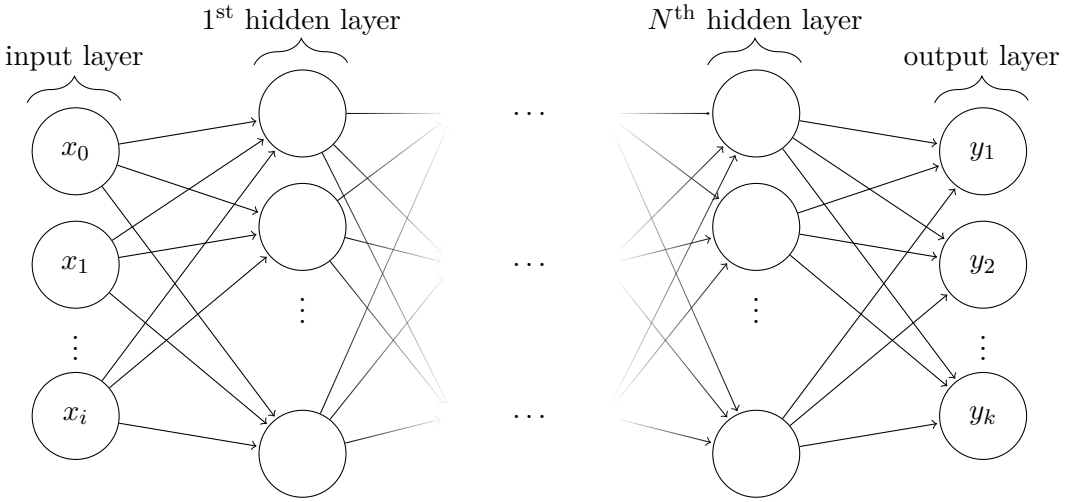


Figure 2.5.: Multi-layer feedforward neural network

2.3.2.5. Long Short-Term Memory

Long Short-Term Memory (LSTM) was first defined by Hochreiter and Schmidhuber (1997), a proposed solution to the vanishing gradient problem. As with typical RNNs, an LSTM NN uses hidden states that it passes along to new cells, along with its *cell state*, c_t. For each cell, four *gates* are created to decide whether to erase a cell (forget-gate), whether to write to a new cell (input-gate), how much to write to a cell (amount-gate) and how much to reveal from a cell (output-gate). Again, as with RNNs, this implementation does not solve the issue of long-term dependencies. An attempt to improve LSTM NNs, especially for text classification, is the Bidirectional LSTM (BiLSTM), where a backward layer is included in addition to the standard forward layer. This improves attention towards the end and start of a document, but still fails to provide insights about the parts in-between.

2.3.2.6. Encoder-Decoder Model

An encoder-decoder model is a two-step process, consisting of an encoder receiving an input sequence, and a decoder producing an output sequence. In Figure 2.6, a simplified example for the task of translating "writing a sentence" into Norwegian is shown. Here, the input sentence is first split into each of its constituent words, then the word vector of each respective word is processed by a RNN, named the encoder, where the states are passed on to the next cell in the network (as described in Section 2.3.2.4). The final encoded hidden state will then be sent through another set of RNNs, named the decoders. In the decoder, necessary techniques are applied to complete the process of translating each word (in context of its hidden state). Finally, it returns the output sequence.

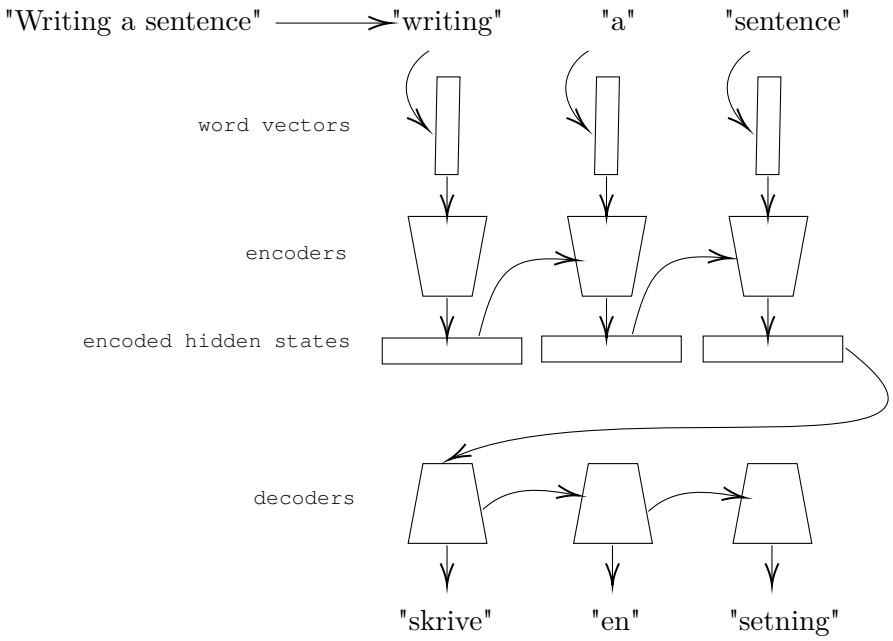


Figure 2.6.: A simplification of an encoder-decoder pattern showing the procedure of translating "Writing a sentence" into Norwegian

2.3.2.7. The Attention Mechanism

Developed by Vaswani et al. (2017), the attention mechanism is inspired by the encoder-decoder pattern, consisting of encoder- and decoder layers. An input is passed into the encoder layer, where each of the encoder outputs is passed into *all* decoder inputs. Each of the encoder layers consists of self-attention and a feedforward NN. Self-attention is the task of assigning which part of a document is related to another part of the same document. Self-attention is computed using three matrices, Q (queries), K (keys) and V (values), in addition to the input itself, which consists of an embedding vector for each term of a document (e.g. word embedding), as well as a positional encoding vector (storing the position of a term). The position of each relevant segment can thus be passed along the encoder layer and decoder layer, without the need of any sequential operations like those found in RNNs, resulting in a fully parallelizable process.

2.3.2.8. Hierarchical Attention Networks

Mirroring the hierarchical structure of documents (paragraphs, sentences, words), a hierarchical attention network (HAN) considers certain parts of a hierarchy based on knowledge about the structure of a document. A distinguished feature of HAN models is that they assign attention weights based on the context-dependence of words and sentences in documents. The sentence "This soda tastes super good" can be represented as "[This soda tastes super][**good**]" (bold part representing word-level attention). Unlike

the attention mechanism described in Section 2.3.2.7, attention is computed using word embeddings (see Section 2.1.4) and aggregating the representation of informative segments that form a document. If a set of unrelated words are combined in a sentence, HAN models avoid assigning high attention scores to these. The sentence "Piano potato roof" has seemingly no attentive words (especially due to its lack of composition). The same goes for subsequent sentences; "My name is" will likely be followed by a name, and not "door". The attention towards "door" would therefore be low, while the attention towards "John" would be higher.

2.3.3. Pre-training

The process of pre-training generally involves training a NN on large-scale, unlabeled text data (Yang et al., 2019b), creating an unsupervised general purpose Language Model (LM), covered in Section 2.1.3. The LM is later specialized by fine-tuning it on domain-specific – or *downstream* – tasks. Downstream tasks (such as question answering, machine translation, reading comprehension and summarization) make use of supervised learning techniques (Radford et al., 2019), thus creating *semi*-supervised classifiers. For language understanding tasks, completely generalized LMs have been implemented (Radford et al., 2019), omitting the use of supervised specialization, relying on detecting the syntagmatic and paradigmatic associations between words. Two widely used LMs (and also Sesame Street characters) are ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019).

ELMo

ELMo (**E**mbeddings from **L**anguage **Mo**dels) create word vectors to model complex word representations. The representations are learned through a bidirectional LM, trained on a large corpus developed by Chelba et al. (2013). At the time of publishing, the implementation of ELMo presented state-of-the-art performance on several NLP tasks, but were later that year outclassed by the introduction of BERT.

BERT

BERT, abbreviated from **Bi**directional **E**ncoder **R**epresentations from **T**ransformers, has redefined performance of several fields within Natural Language Processing in the past year (Radford et al., 2019). Its functionality is dependent on, as the name suggests, *transformers*. Keep in mind the following information is aimed to describe BERT on a high level. An excellent, more in-depth description can be found in the Master's Thesis by Steinbakken (2019), in addition to the source papers on Attention (Vaswani et al., 2017) and BERT (Devlin et al., 2019).

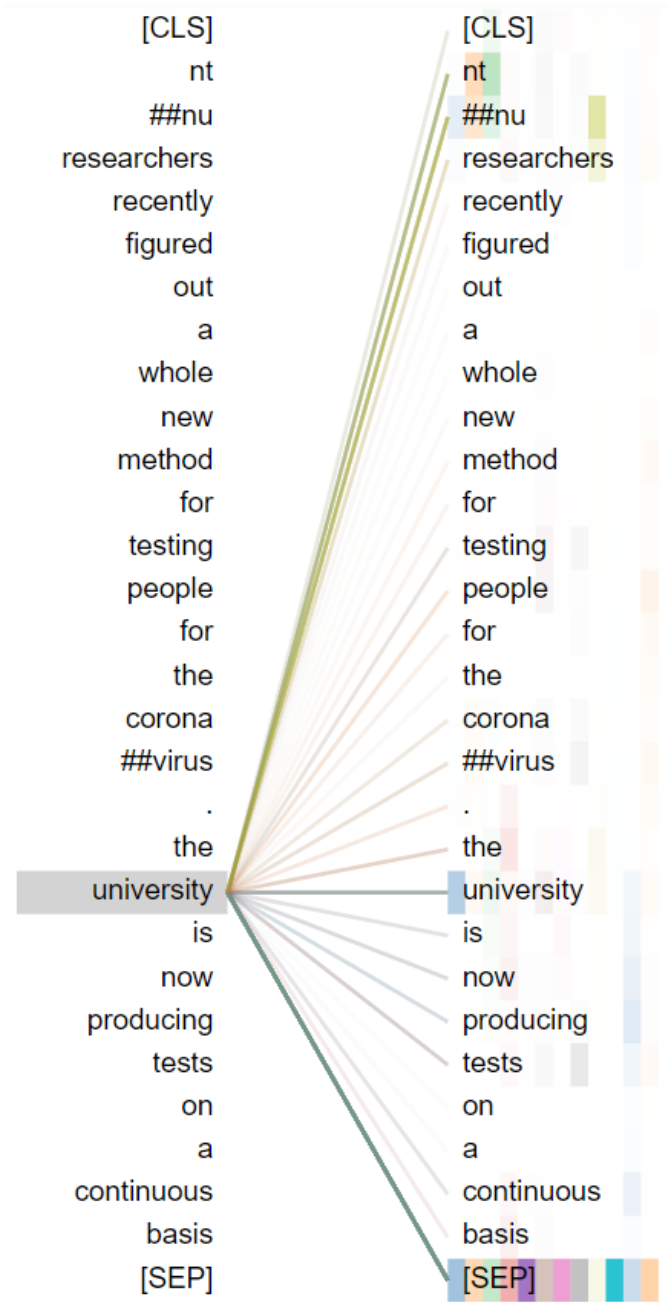


Figure 2.7.: Example visualization of an output from a BERT model

Transformers Transformers incorporate attention mechanisms (described in Section 2.3.2.7, p. 19), applying self-attention mechanisms and modeling the relationships between tokens (i.e. words) in sentences without regard for the positional information of the respective tokens, but keeping track of its *direction* (Vaswani et al., 2017).

This differs from the functionality of alternative neural architectures, which often encode positional information (e.g. bidirectional LSTM). The positional independence of tokens allow for a greater understanding of language, as ambiguous words are embedded with their respective context in any sentence – allowing the model to discern if "run" refers to a marathon or a horse race, depending on its contexts from other words in the sentence. The final representation of the word "run" contains attention scores (i.e. self-attention to "run") for any other word, in relation to itself. The keen reader may see that this approach will quickly result in large models, as each token in a text effectively contains a copy of the text itself (represented as vectors).

Using the transformer technique, BERT operates by encoding bidirectional transformers, i.e. jointly applying self-attention both left and right (Devlin et al., 2019), learning intricate relationships between tokens in a text. By pre-training BERT on large-scale text data, requiring access to excessive hardware, the model may be redistributed for others to use, without the need for additional training. Additionally, the models can be fine-tuned, as described earlier, and these fine-tuned models may also be distributed and modified for the convenience of end users. An illustration of BERT in presented in Figure 2.7, where the different colors indicate attention towards tokens. The *[CLS]* token indicates *start of classification* and the *[SEP]*, or *separator*, indicates when to separate between classifications. Note how it deals with unknown words, such as the "coronavirus", being split as "corona" and "##virus" and the same for "NTNU". This allows the model to map "virus" to any previously seen occurrences of virus in a text. Observe in the figure how the attention towards *nt* and *##nu* is prominent. This is the power of pre-training, as it has discovered this pattern between how any unknown token (e.g. "##nu") is very plausibly referring to "the university" based on its encoded directions.

2.3.4. Configuring Machine Learning Classifiers

As a last section on machine learning classifiers, some explanations on the jargon used when configuring these classifiers is presented.

Hyperparameters A model's parameters – hyperparameters – refer to any parameter set before the learning process takes place. The purpose of most parameters is to enhance the chances of the model learning adequately from the data, avoiding too close approximations (overfitting) or too loose approximations (underfitting). The latter may also occur when there is a lack of data necessary to learn intrinsic patterns in the data. Some common hyperparameters include:

- **Epochs**
  An epoch is the term used to define a single pass of the dataset through the machine learning model. Several epochs are often required for the model to approach the global minimum with respect to the loss function. Too few epochs can result in the model stopping before it has reached the point of convergence.

- **Batch size**
  The batch size is the number of objects to include in a single batch, where a batch is a predefined portion of the dataset. A low batch size, e.g. 1, implies that the model learns from a single text at a time, whereas a larger batch size will cause the model to learn more complex structures as all the inputs are handled at once. If a dataset is split into 10 batches, it will consequently require 10 iterations to finish one epoch of training.
- **Learning rate**
  This parameter changes how much the model learns from its input data. High learning rates require less epochs to find a solution (although the solution may be suboptimal), whereas a low learning rate may not be able to find a solution at all. Thus, a middle ground has to be defined.
- **Dropout**
  A dropout has the functionality of dropping, or ignoring the outputs from a given number of neurons in a network, typically at random with a defined probability. A dropout of 1.0 would disregard all outputs, not allowing the model to learn at all. With a dropout at 0.0, all neurons would be passed along the layers in the networks, which may in turn cause overfitting.
- **L2-regularization**
  Regularization works by adjusting how the loss function impacts the complexity of the model. The weights at each neuron are forced to become small – depending on the value of the L2-regularization – resulting in the model being less likely to latch on to discovered patterns. This further prevents overfitting.

Overfitting Overfitting happens when the model is too closely fit to the data. If a model is strictly trained on data from a specific topic, for instance, it may perform poorly on other, never before seen topics.

2.4. Evaluation Metrics

Throughout the thesis, several evaluation metrics will be mentioned when discussing performance, as well as in grouped results in tables and by other means. First, general evaluation metrics for classification (e.g. Sentiment Analysis) are described, before moving on to more specialized metrics for Coreference Resolution (CR).

2.4.1. Sentiment Analysis

Most commonly used is the F_1-score, derived from *precision* and *recall*. Another common evaluation is *accuracy*, an intuitive score which is the fraction of correct predictions made out of all predictions. The evaluation metrics make use of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) to describe the outcomes of a prediction, illustrated by the confusion matrix in table 2.3.

		Predicted values: Positive	Predicted values: Negative
True values	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 2.3.: Confusion matrix for prediction outcomes

Precision

Proportion of returned items that are relevant:

$$precision = \frac{TP}{TP + FP}$$

Recall

Proportion of relevant items that are returned:

$$recall = \frac{TP}{TP + FN}$$

F-score

The F-score, is a combination of precision and recall, weighted with a variable β:

$$F_\beta = (1 + \beta) * \frac{precision * recall}{\beta^2 * precision + recall}$$

The F_1-score ($\beta = 1$) is the harmonic mean of precision and recall.

$$F_1 = (1 + 1) * \frac{precision * recall}{1^2 * precision + recall} = 2 * \frac{precision * recall}{precision + recall}$$

This weighted metric is commonly used in text classification, as very precise models (i.e. high precision) will be penalized by not retrieving a high number of relevant items to make the prediction on (i.e. recall).

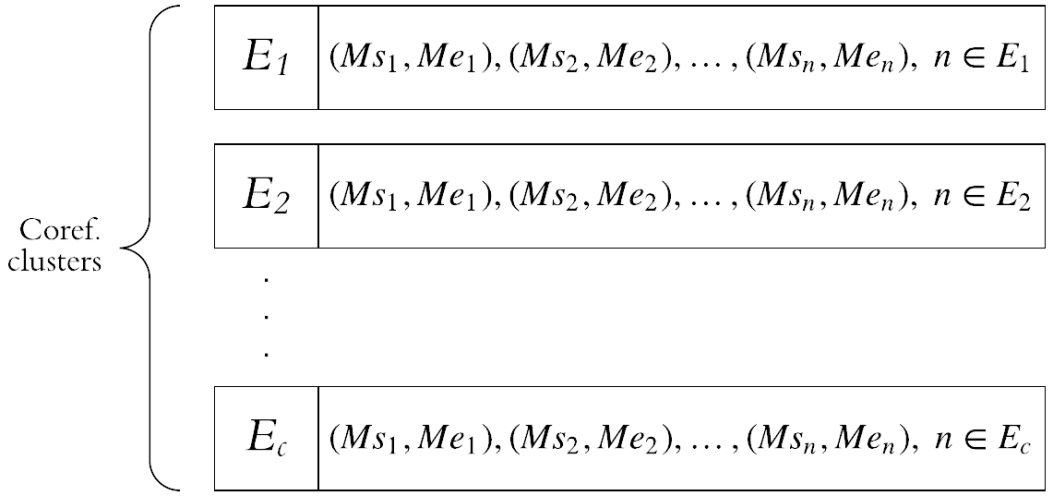


Figure 2.8.: Data structure holding coreference information for entities

2.4.2. Coreference Resolution

The previous metrics do not apply well to CR, as this task is not a traditional classification task. The definitions of precision and recall thus require modification to suit the desired outcome in evaluating references to entities. There are a vast set of metrics available, but the most commonly used are described below. Before introducing the metrics, it is favorable to understand the input of how entities and their coreferences are represented (although the data structures alter between models – a topic that will later be discussed). The chosen data structure contains a list of lists of tuples. First, the main list contains the coreference clusters (C). Within C are a list of entities ($E_1, E_2, \ldots, E_c,\ c \in C$) for the given cluster, each containing a list of mentions $M_1, M_2, \ldots, M_n,\ n \in E_c, c \in C$). The mention lists contain mention tuples with their respective start and end indices, denoted Ms and Me. This structure is illustrated in Figure 2.8. Furthermore, when referring to the truth values of test data, the term *gold* is used, as it is the standard terminology in the field. For mathematical notations, gold data is denoted G and predictions P, containing mentions for entity clusters. Consider the text from Example 1 in the Introduction (Chapter 1): "Anna bought John a bike. He told her it was great!". The gold mentions are as follows: $[[(0,0),(8,8)],[(2,2),(6,6)],[(3,4),(9,9)]]$ Translating to **[Anna, her], [John, He], [a bike, it]**. This example will be used to illustrate the different metrics. Additionally, the formulas shown below are based on the study by Moosavi and Strube (2016) and verified against the metric source papers and implementations by Lee et al. (2018). All metrics are used for evaluations throughout the experimental sections.

MUC

Vilain et al. (1995) introduced MUC, the very first CR metric, used for evaluation in the *Message Understanding Conference* (Grishman and Sundheim, 1996). MUC only considers the difference between coreference links against the gold entities – in other words, calculating the number of changes required to recreate the gold mentions from the predictions, much alike the functionality of Levenshtein distance (Levenshtein, 1966). Recall is computed by iterating over the correct mentions in the predictions for each entity, computing its intersection with the gold mentions, using a function *partition* (Equation 2.1), partitioning the input clusters that intersect.

$$partition(x, y) = \{x | x \in X \& x \in y \neq \emptyset\} \tag{2.1}$$

$$Recall_{MUC}(P, G) = \sum_{g \in G} \frac{|g| - |partition(g, P)|}{|g| - 1} \tag{2.2}$$

Precision is calculated by simply swapping the inputs P and G. This simplistic comparison leaves it unable to differentiate between singleton mentions and references between entities. Thus, the evaluation scores will still be high if all coreference links are merged into one single entity, as illustrated below. Observe how the predictions are all in one list [**Anna, her, John, He, a bike, it**]:

```
Predicted: [((0, 0), (8, 8), (2, 2), (6, 6), (3, 4), (9, 9))]
Gold: [((0, 0), (8, 8)), ((2, 2), (6, 6)), ((3, 4), (9, 9))]
Running metric: muc
Precision:      0.6
Recall:         1.0
F1 score:       0.7499999999999999
```

B-CUBED

B^3 (Bagga and Baldwin, 1998) addressed some issues of MUC, by introducing mention-based metrics – a metric that computes the recall and precision from individual entity mentions, rather than as a whole, before averaging for the final scores.

$$\text{Recall}_{B^3}(P, G) = \frac{\sum_{e_g \in G} \sum_{e_p \in P} \frac{|e_g \cap e_p|^2}{|e_g|}}{\sum_{e_g \in G} |e_g|} \tag{2.3}$$

As with MUC, precision is computed by swapping the inputs.

CEAF

Luo (2005) developed the CEAF metric to address issues with previous metrics. A predicted entity should only map to one gold entity. A similarity is computed between the predictions and gold data, where the task is to find a one-to-one mapping using the

Hungarian Algorithm (Kuhn, 1955). Its formulas are unnecessarily complex to justify listing here, and a detailed description is found in Moosavi and Strube (2016). Its main takeaway is the one-to-one mapping, penalizing the addition of wrongly identified coreference clusters, illustrated with the addition of **[Anna, John]** as a cluster:

```
Predicted: [((0, 0), (2, 2)), ((0, 0), (8, 8)), ((2, 2), (6, 6)
   ), ((3, 4), (9, 9))]
Gold: [((0, 0), (8, 8)), ((2, 2), (6, 6)), ((3, 4), (9, 9))]

Running metric: b_cubed
Precision:      0.875
Recall:         1.0
F1 score:       0.9333333333333333
-----------------------------------
Running metric: ceaf
Precision:      0.75
Recall:         1.0
F1 score:       0.8571428571428571
```

Observe how the CEAF score is lower than that of B-CUBED after this incorrect prediction.

CoNLL-2012

The CoNLL metric has been used in almost all work done on CR since the CoNLL-2012 shared task (Pradhan et al., 2012). It is quite simply an average of MUC, B^3 and CEAF, smoothing out the shortcomings of each one. Unless specified otherwise, the F_1 scores

LEA

Moosavi and Strube (2016) discussed issues with the evaluation scheme of the CoNLL-2012 shared task and its metrics, and proposed the LEA – Link-based Entity-Aware – metric. It considers how well each entity is resolved, by computing the fraction of correctly predicted coreference links. Thus, the more links, the higher an entity is scored. This metric is intended to be used where the importance of entities should be evaluated. The authors further suggest modifying the importance score (Equation 2.4) based on domain-specific needs.

$$\mathit{importance}(e) = |e| \tag{2.4}$$

$$\mathit{link}(e) = |e_{\mathit{mentions}}| \times \frac{|e_{\mathit{mentions}}| - 1}{2} \tag{2.5}$$

$$\mathit{resolution\text{-}score}(e_g) = \sum_{e_p \in P} \frac{\mathit{link}(e_g \cap e_p)}{\mathit{link}(e_g)} \tag{2.6}$$

The evaluation of entities is illustrated in the following equation (2.7). The more complex equations for recall and precision provide little needed information to convey here, and can be found in Moosavi and Strube (2016).

$$\frac{\sum_{e_p \in P} (importance(e_p) \times resolution\text{-}score(e_p))}{\sum_{e_g \in G} importance(e_g)} \tag{2.7}$$

2.5. Tools

This section is dedicated to providing a brief description of tools, frameworks and libraries used in the implementation of experiments.

2.5.1. GraphQL

GraphQL is a query language for application programming interfaces, developed by Facebook. In this thesis, it is used to specify data from the Strise Knowledge Graph. GraphQL allows for customizable data retrieval description, specifically made for graphs, as the name indicates. More can be read in the analysis of the language by Hartig (2017).

2.5.2. Python and Related Tools

Below are some tools used in relation to the Python programming language.

Pandas

Pandas is a data analysis tool for Python (McKinney et al., 2011), allowing efficient creation of large objects with several built-in methods for manipulating data.

Numpy

A package for Python to perform efficient computing. Heavily used in a vast majority of modern machine learning systems. It will be mentioned throughout the Architecture Chapter (Section 5), more specifically the *array* functionality – N-dimensional objects optimized for fast calculations.

NLTK

NLTK, the Natural Language ToolKit (Loper and Bird, 2002), is a tookit providing access to tools like tokenization and dependency tree visualization, as well as high-level interfaces to semantic parsing, machine translation and sentiment analysis.

spaCy

spaCy (Honnibal and Montani, 2017) is an open-source library for a plethora of NLP tasks. It supports high-level APIs for (world's fastest) dependency parser (Choi et al., 2015), tokenization, Named Entity Recognition (NER) and more. It allows for modules to be built on top of its system, creating custom pipelines.

NeuralCoref

NeuralCoref is a system developed for solving CR by Hugging Face[3], a company providing open-source tools for several tasks in NLP. It supports integration with spaCy, which will be utilized to build an efficient pipeline for evaluating coreferences. The system is based on the algorithms by Clark and Manning (2016a), and further modified for efficiency.

PyTorch

PyTorch (Paszke et al., 2019) is a library for enabling development of Deep Learning applications in Python. It is the main library for models regarding sentiment classification in this thesis.

Jupyter Notebook

Jupyter provides a *notebook* service, which allows to efficiently split Python code into individual blocks of code, storing the blocks in memory. Moreover, it allows for modules to be applied, supporting advanced graphics, plots and HTML views.

[3] `https://huggingface.co/`

3. Related Work

This chapter contains methodologies and compelling approaches used to carry out experiments for the rest of the thesis. A structured, domain oriented literature review is first presented, followed by a categorized listing of related work based on the results in later sections.

3.1. Literature Review

The goal of the literature review is to obtain sufficient knowledge within the field of Coreference Resolution (CR) and discover high-performing, intriguing models, with focus on the reproducibility of results. The review was initially guided by findings made in the specialization project (Jørgensen, 2019), wherein several fields of Natural Language Processing (NLP), e.g. CR, were researched in an extended topic prestudy. Furthermore, state-of-the-art methods used across many NLP tasks were studied, such as BERT (Devlin et al., 2019), a language model based on pre-training (covered in Section 2.3.3). Starting the specified study of CR, a domain oriented review protocol is defined (Section 3.1.1), including the selection and quality assessment of research. Finally, an overview of studied literature is presented, followed by some brief remarks.

3.1.1. Domain Oriented Review Protocol

Initiating the search for publications, terms related to the topic must be defined. Due to the extensive review completed in the specialization project, in which part of the research goal was to "get a clear view of the state-of-the-art techniques in the field of Natural Language Processing (NLP)", several key terms in the topic of CR were already

Topic	**Returned search results**
Coreference resolution	13,500
Named entity recognition	149,000
Mention detection	221,000
Entity linking	309,000
Sentiment analysis	714,000
Machine translation	1,420,000

Table 3.1.: Number of returned search results (from the years 2010–2020) on Google Scholar for several related topics

	Group 1 **Main topic**	**Group 2** **Specialized terms**	**Group 3** **Linguistic terms**	**Group 4** **External data**
Term 1	Coreference resolution	Entity	Anaphora	Knowledge base
Term 2	Co-reference resolution	Entity-level	Anaphoric	Knowledge bases
Term 3	-	Span	Cataphora	Knowledge-base
Term 4	-	Spans	Cataphoric	Knowledge driven
Term 5	-	Span-level	Antecedent	Knowledge-driven
Term 6	-	-	Coreferring	-

Table 3.2.: Terms used for the literature review

discovered. Continued, a broad online search was conducted in order to find terms to be used in the literature review. An enormous gap between the amount of published research of CR versus other areas in NLP was quickly discovered, as shown in table 3.1.

The final terms (Table 3.2) were used to create a boolean search query for Google Scholar with the following scheme:

```
Group = Term1 OR Term2 OR ... OR TermN
Query = Group1 AND Group2 AND ... AND GroupN
```

Due to query length restrictions of 255 characters per search, the four groups could not be combined into one single query, and were thus split on the least essential terms into the following two queries:

Query 1 ($Q1$)

```
("coreference resolution" OR "co-reference resolution")
AND
("entity" OR "entity-level" OR "span" OR "spans" OR "span-level")
AND
("knowledge base" OR "knowledge bases" OR "knowledge-base"
OR "knowledge driven" OR "knowledge-driven")
```

Query 2 ($Q2$)

```
("coreference resolution" OR "co-reference resolution")
AND
("entity" OR "entity-level" OR "span" OR "spans" OR "span-level")
AND
("anaphora" OR "anaphoric" OR "cataphora" OR "cataphoric"
OR "antecedent" OR "coreferring")
```

3.1.2. Restricting the Search Scope

Before applying the two queries to conduct a search on Google Scholar, a vital decision had to be made – restricting the search within a given time frame. Guided by NLP-

progress[1], an open-source project by Sebastian Ruder, a Deepmind[2] research scientist, it was found that the performance of recent CR systems drastically increased in the past two years. This, combined with findings from the specialization project, resulted in selecting the time frame 2018–2020 to acquire the most up-to-date and relevant research. Additionally, it was decided to keep a total of 30 publications after filtering on inclusion criteria. This decision was motivated by previous experiences with literature reviews, as publications of high quality often refer to other relevant publications – drastically increasing the amount of material for the entire review.

3.1.3. Selection of Studies

For retrieved publications, each was respectively asserted by two inclusion criteria (IC). These mainly regard the title, abstract and results of a publication, and are merely binary assertions to decide whether to continue studying a given publication or not.

IC1 Is Coreference Resolution the main topic of the research?

IC2 Are relevant results presented?

When inspecting the results throughout this inclusion process, 13 publications were kept from $Q1$ (Appendix A.1) and 17 from $Q2$ (Appendix A.2), yielding 30 publications to be further processed by the quality assessment.

3.1.4. Quality Assessment

A set of quality criteria (QC) were specially crafted for this thesis, to properly attribute scores and rank retrieved publications:

QC1 Is the goal of the research clearly stated, including its relation to coreference?

QC2 Are the results based on available datasets?

QC3 Are the results compared to other relevant research?

QC4 Is the experimental setup reproducible, with public code?

QC5 Is external data, such as knowledge bases, considered in the research?

QC6 Is the proposed solution showing state-of-the-art performance?

[1] http://nlpprogress.com/
[2] https://deepmind.com/

3.1.5. Review Workflow

A publication received up to 1 point for each QC if answered, 0.5 for partial answers and 0 otherwise. For a publication to be included in the final review library (Appendix A, Table A.5), it must score at least 4 out of 6 available points. The evaluation process was completed by the following workflow:

1. (QC1) Consider the title and read abstract
2. (QC1, QC2) Read introduction
3. (QC2) Search for datasets used
4. (QC3, QC6) Consider the results, check for similarities in other papers
5. (QC4, QC5) Skim through experimental setup and conclusion, look for available code and other means of reproducibility

The attributed scores are shown in Appendix A, Tables A.2 ($Q1$) and A.4 ($Q2$), in which the ID column is linked to the previous tables. The publications passing the quality assessment were gathered in a final review library (Appendix A, Table A.5). The continued review process was then bisected:

1. Get an overview of the field: read reviews and surveys
2. Delve into deeper details: study of remaining publications

3.1.6. Results

Sukthanker et al. (2018) and Stylianou and Vlahavas (2019) provide a good overview of CR, comparing a plethora of models and their evolution from simple rule-based design up to and including the current state-of-the-art neural architectures. The following sections present findings from the literature review, describing algorithms for the CR task, using world knowledge for CR and applications to Sentiment Analysis (SA).

Factor Type	Initial Weight
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect object and oblique complement emphasis	40
Head noun emphasis	80
Non-adverbial emphasis	50

Table 3.3.: Salience Factor Types as defined by Lappin and Leass (1994)

3.2. Algorithms for Coreference Resolution

Algorithms used to handle CR are divided into rule-based and supervised approaches. Historically, the CR task has required some form of hand-crafted rules (Sukthanker et al., 2018), containing carefully extracted syntactic and semantic features. Which features, however, is an on-going discussion (Bengtson and Roth, 2008; Moosavi and Strube, 2017). Below, a brief explanation of the most used algorithms are described.

3.2.1. Rule-based Algorithms

The very first documented approach to a CR related task, namely pronoun resolution, was developed by Hobbs (1978). Hobb's naïve algorithm describes a traversal of parse trees of sentences in a left-to-right manner using breadth-first searches to find an antecedent for pronouns. Lappin and Leass (1994) developed an algorithm that implemented a look-back search for sentences, removing mention candidates that did not semantically or syntactically agree. Furthermore, they defined a set of salience factor types with respective weights to extract the remaining candidates, shown in Table 3.3.

Fast-forward to modern times, Stanford CoreNLP (Manning et al., 2014a), an NLP toolkit, is widely used for several NLP tasks. Its coreference models include a deterministic, rule-based system implemented by Lee et al. (2013), with later additions of statistical and neural models by Clark and Manning (2015) and Clark and Manning (2016a). The rule-based system makes use of a multi-pass sieve for CR. A simplified description of the sieve-based architecture (Lee et al., 2013) can be found in Table 3.4. The modularity of the system allows for simple integration of other techniques. Lee et al. (2013) conclude with shallow knowledge of semantics and discourse to be the main cause of errors. Additional approaches to other rule-based algorithms are thoroughly covered by Sukthanker et al. (2018). A common problem with most rule-based algorithms is that references in natural languages are largely disordered, and thus cannot be completely solved in a left-to-right manner, or any sequential manner, in fact, but should rather be handled as a hierarchical problem. This is where, among several issues, supervised algorithms may excel.

Sieve	Description
Input	Raw input data
Mention detection	Discover entities, pronouns, and so forth
Speaker identification	Match words like “my”, “mine”, based on the speaker. The speaker can be inferred by connected entities and pronouns, largely based on the part-of-speech
String match	Match exact strings, like “NTNU” can safely be matched with a later occurrence of “NTNU”
Precise constructs	Match syntactic constructs, often grouped relations. For instance, “her best friend”
Strict head match	A total of three sieves, denoted A, B, C in the publication. These match the root antecedent of previous matches
Proper head noun match	Looks for certain constraints on head matches before resolving head match candidates, such as location and numeric matches.
Relaxed head match	Matches part of words, such as “University” to “Norwegian University of Science and Technology”
Pronoun match	Implementations of agreement constraints to enforce validity of number, gender, animacy, and so forth
Post processing	Corpus-specific post processing techniques. Was only implemented for the OntoNotes dataset

Table 3.4.: Sieve-based architecture for the deterministic model by Lee et al. (2013)

3.2.2. Supervised Algorithms

Some of the earlier learning-based models made use of decision trees (Aone and Bennett, 1995; Soon et al., 2001) and the bayesian rule (Ge et al., 1998). These were popularized at the time due to the work on tagged corpora like the MUC-6 (Grishman and Sundheim, 1996). Although supervised methods have in recent years remained superior over rule-based, Zeldes and Zhang (2016) show that supervised models are weaker when evaluated on intricate linguistic phenomena within coreference and mention-border definitions. This is due to the fact that a single dataset will never be sufficient for a model to learn all nuances within a language from, as one cannot feasibly represent ever-changing natural languages in a compact selection of data.

Supervised approaches may be separated into three categories:

- Mention-pairs
- Entity-mentions
- Ranking models

3.2.2.1. Mention-pair models

Mention-pair models have the goal of classifying whether pairs of noun phrases (NPs) are coreferent, as stated in the previously mentioned work by Aone and Bennett (1995). Sukthanker et al. (2018) split the algorithmic design of mention-pair models into:

1. Instance creation between two NPs
2. Training a classifier on the instances
3. Generating NP partitions by clustering

Early work on mention-pair algorithms, such as the right-to-left clustering by Ng and Cardie (2002), has been used (although slightly modified) in modern models (Lee et al., 2017). One major flaw of the mention-pair modeling scheme is that transitive relations do not uphold if there exists misclassifications in one of the relations. This is illustrated by the following example: given an entity $\mathcal{E}$ with the following NPs as references: $\mathcal{S} = \{A, B, C\}$. A mention-pair model will attempt to link the NPs in the set $\mathcal{S}$ to $\mathcal{E}$. The following behavior is ideal (note the right-to-left ordering):

$$(C \to B) \wedge (B \to A) \wedge (A \to \mathcal{E}) \implies \forall x \in \mathcal{S} : x \to \mathcal{E}$$

If, however, any link in the set is unresolved, such as $(B \not\to A)$, the outcome is:

$$(C \to B) \wedge (B \not\to A) \wedge (A \to \mathcal{E}) \implies A \to \mathcal{E}$$

Thus, the first two links are missing due to one missing reference. This problem will only become more severe in longer chains of mention-pair links.

3.2.2.2. Entity-mention models

Given a singular entity, entity-mention models attempt to classify whether NPs are coreferent with previously established clusters of entities and coreferences, as opposed to a single antecedent. Defining cluster-level features to aid in the creation of datasets proved extremely difficult in earlier models (Sukthanker et al., 2018), although later made accessible by neural models. However, the entity-mention modeling approach had proven to be subpar compared to mention-pair algorithms.

3.2.2.3. Ranking models

With modern machine learning approaches, previous outcomes from binary models (such as the mention-pair) were modified to a regression problem, determining how good a given antecedent was in comparison to other antecedents for any given reference. In fact, Hobb's algorithm (Hobbs, 1978) could be seen as a ranking model, as given phrases were passed down a sieve of constraints in order to find the highest ranked antecedent. Ranking models by Björkelund and Farkas (2012) and Durrett and Klein (2013) excelled in the shared task CoNLL-2011 (Pradhan et al., 2011) and the successive CoNLL-2012 (Pradhan

et al., 2012). Although ranking models seemed to perform well, the introduction of deep learning models seemed to better capture the complex structure of CR, as evident by the performance of Clark and Manning (2016a), surpassing all scores for the CoNLL-2012 task (Pradhan et al., 2012) at the time. Further performance on the CoNLL-2012 task will be documented throughout the next section on more advanced models, when exploring deep learning and Neural Network (NN)s. At the time of closing the shared task[3], the models by Fernandes et al. (2012) and Björkelund and Farkas (2012) excelled. Table 3.5 shows their respective CoNNL F1 scores.

Author	**CoNLL-2012 F1 Score**
Fernandes et al. (2012)	63.37
Björkelund and Farkas (2012)	61.24

Table 3.5.: Two top performing models at the closing of the CoNLL-2012 shared task.

3.2.3. Deep Learning and Neural Networks

Due to the meticulous work required to create and update features by hand (e.g. those found in rule-based models), being able to circumvent this task is more efficient (with respect to manual labor) and possibly less prone to human errors. Deep learning models for NLP and CR were severely enhanced by using word embeddings such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), techniques that represent words as vectors with several features, according to their semantic properties. Up until 2015, there was little work to be found on deep learning and neural architectures for the CR task. The earliest NN based models developed by Wiseman et al. (2015), Wiseman et al. (2016), Clark and Manning (2016b) and Clark and Manning (2016a) slightly beat the previous best ranking models (see Table 3.5). These models handled specialized subtasks such as anaphoricity detection and antecedent ranking (Wiseman et al., 2015), as well as learning global information from entity clusters without predefining cluster features (stated as a previous issue in Section 3.2.2.2) (Wiseman et al., 2016). Clark and Manning (2016b) trained a model to distinguish clusters of related mentions, as well as testing the benefits of hand-engineered features implemented in the final model, where the observations are listed in Table 3.6. The features in this model diverge somewhat from those originally used by Lappin and Leass (1994), and combining these, or doing a feature-importance search on a broader set of features could have its merits. In Clark and Manning (2016a), the first high-performing reinforcement learning model was implemented, optimizing a mention-ranking model for common evaluation metrics. This approach eliminates the need for tedious hyperparameter optimization.

Lee et al. (2017) introduced the first end-to-end CR model, outperforming all previous models at the time, discovering that hand-labeled features are not necessarily required in order to further improve neural models for CR. The high-level functionality of the model

[3] http://conll.cemantix.org/2012/

Feature	Impact on F1 score
Mention	1.27
Document genre	0.25
Distance	2.42
Speaker	1.26
String matching	2.07
Total	7.27

Table 3.6.: Hand-engineered feature contribution by Clark and Manning (2016b)

is to jointly learn spans correlating to entity mentions and how to cluster the mentions. The same authors later published a higher-order CR system (Lee et al., 2018), utilizing the antecedent distribution output from a span-ranking architecture as an attention mechanism. The attention mechanism continuously refined the span representations. This system was developed as previous models (Clark and Manning, 2016a; Lee et al., 2017) were prone to predict globally inconsistent features, as only local contexts were modeled. The model is built upon that of Lee et al. (2017) with ELMo representations (Peters et al., 2018), hyperparameter tuning, course-to-fine and second-order inference – each step slightly increasing the overall performance. Subramanian and Roth (2019) found generalization issues in the Lee et al. (2018) model and aimed at improving the training of neural CR systems. Generalization issues in the commonly used datasets have previously been addressed by Moosavi and Strube (2018) – further examined in Moosavi (2020). By retraining the model by Lee et al. (2018) with adversarial training, Subramanian and Roth (2019) further improved the scores on the CoNLL-2012 task and GAP dataset. Fei et al. (2019) developed a goal-directed end-to-end reinforcement learning model, once again built upon the model by Lee et al. (2018) and slightly improved its performance.

3.2.4. Pre-training

Another cluster-centric approach by Kantor and Globerson (2019) built upon the cluster-level features enabled by NNs, as presented in Wiseman et al. (2016). The model makes use of BERT embeddings (Devlin et al., 2019) to achieve an F1 score improvement of 3.6 percent points on the CoNLL-2012 task. The approach focuses on entity-level information, as they saw issues with previous work mostly depending on pairwise scoring of entity mentions (Kantor and Globerson, 2019) – missing global entity information. The underlying hypothesis of the implementation is that "each entity should be represented via the sum of its corresponding mention representations" (Kantor and Globerson, 2019).

Joshi et al. (2019b) also experimented with the usage of BERT for the CR task. By modifying the model proposed by Lee et al. (2018) and replacing the ELMo representations with BERT transformers, they achieved slight improvements over the model by Kantor and Globerson (2019). Another, later, model by the same authors implements span prediction with BERT (Joshi et al., 2019a). SpanBERT, like BERT, is not a specific

implementation for the CR task, but rather a language model that may be fine-tuned on downstream tasks (among those, CR). The main difference between the BERT and SpanBERT models is that BERT randomly masks tokens of text, whereas SpanBERT masks spans of text. The length of the span is randomly selected from a geometric distribution (ranging from 1 to 10), and experiments show that this approach forces the model to better learn textual properties. Finally, Wu et al. (2019) state some concerns with previous implementations of CR models (Lee et al., 2017, 2018; Zhang et al., 2018; Kantor and Globerson, 2019), in which all text spans in a document are considered potential mentions, and the goal is to attribute an antecedent for the mentions. When these models attempt to connect a mention with an antecedent in a span, they may wrongfully miss certain mentions (as no models are perfect). This leaves out possible mentions in a text span, which will never be processed further. By reformulating the CR task as a Machine Reading Comprehension (MRC) query-based task, Wu et al. (2019) reduce the effect of undetected mentions in the mention-detection phase of a CR system. Additionally, when handling CR as a query-based task, new datasets in the domain of question-answering can be used, possibly increasing generalization of the models. Their model utilizes SpanBERT (Joshi et al., 2019a), and has the current highest evaluated performance on the CoNNL-2012 task, with an F1 score of 83.1.

The results from all the models mentioned throughout this chapter on the CoNLL-2012 task are presented in Section 3.5.

3.3. Incorporating World Knowledge

The need for world knowledge in CR has been known since the nineties (Sukthanker et al., 2018), but little research has been done on the field. Rahman and Ng (2011) implemented world knowledge in a CR system, using external sources for external data, e.g. YAGO and FrameNet. Evaluating on the OntoNotes dataset, it resulted in improvements of up to 4.1 percent points over a baseline model. Uryupina et al. (2011) made use of both YAGO and Wikipedia data with around 2 percent points improvements over a baseline model. These results definitely show that making use of knowledge bases is of interest, although the results are minor. Zhang et al. (2019) successfully incorporated world knowledge in a pronoun-specific CR model, and is the first work to do so for deep learning models. This model is based around an attention mechanism to correctly deduce *which* knowledge to use in a given context, as first discussed (but not implemented) by Lee et al. (2017). As the proposed solution is specialized towards pronouns, it is thus compared to a baseline with other similar benchmarks (third personal and possessive pronouns). Their model resulted in substantial improvements over Lee et al. (2018), Clark and Manning (2016a) and Clark and Manning (2015) on the same benchmarks.

Author	English CoNLL-2012 F1 Score
Wiseman et al. (2015)	63.39
Wiseman et al. (2016)	64.21
Clark and Manning (2016b)	65.29
Clark and Manning (2016a)	65.73
Lee et al. (2017)	68.80
Lee et al. (2018)	73.00
Fei et al. (2019)	73.80
Subramanian and Roth (2019)	74.70
Kantor and Globerson (2019)	76.61
Joshi et al. (2019b)	76.90
Joshi et al. (2019a)	79.60
Wu et al. (2019)	83.10

Table 3.7.: Neural Network models for the CoNLL-2012 shared task.

3.4. Applying Coreference Resolution to Sentiment Analysis

SA is one of the most popular studies within Natural Language Processing (Liu, 2017), and it is thus surprising to observe the absence of research done on the implementation of CR for SA. A brief study by Nicolov et al. (2008) presented great improvements with CR as an additional layer in the SA pipeline. The study, however, only took local findings into account, due to the lack of commonly distributed baselines at the time. The results can thus not be reproduced. Jakob and Gurevych (2010) speculated that opinion target extraction may benefit by implementing anaphora resolution with an extended version of the CogNIAC algorithm (Baldwin, 1997). The evaluated F1-measures did not see any notable gains, however, as false positives hindered the recall of the actual opinionated documents. Sentiment scopes, a key issue within SA (Liu, 2012), were researched in detail by Li and Lu (2017, 2019) in order to determine a sentiment scope for entities (much alike the process to discover mention candidates for CR). However, CR as a heuristic was completely disregarded in these studies, which resulted in sentiment scopes being generated on a lesser percent of the actual opinionated text. With the introduction of generalized CR models, the issues faced by Jakob and Gurevych (2010) may be resolved.

3.5. Recap and Remarks

Table 3.7 shows the averaged F1 score on the English dataset of the CoNLL-2012 task for publications mentioned throughout the chapter, in chronological order. From the material covered in Sections 3.2 and 3.3, there are a few notes to make on the work done with coreference resolution to date.

3.5.1. Large Neural Architectures and Computing Power

Most recent state-of-the-art models have included implementations of NNs and pre-trained embeddings such as BERT or ELMo (see Background Section 2.3.3 for details on ELMo/BERT). The current best-performing model by Wu et al. (2019) involves a ground-breaking observation regarding the transformation of the CR task to a query-based MRC task. However, the model's complexity is evident (due to its extra layers and datasets involved), taking longer to train compared to other related models – which are already extremely hardware-intensive. Furthermore, the results presented by Wu et al. (2019) are (in its current state[4]) impossible to validate, as core pieces of the evaluation code is strictly missing from the open sourced code-base. Regardless of its validity, the MRC transformation needs to see more research in its implementation before continued as a de facto standard in the field of CR. The SpanBERT model (Joshi et al., 2019a) has so far shown the best results with a typical CR approach, detecting mentions and latent spans in which the references to an entity resides. Both SpanBERT and the original implementation by Lee et al. (2018) are likely to be involved in the path to develop newer models due to their great performance, although they require access to high-performing hardware, mostly limited to research institutions.

3.5.2. Identifying a *Good* Coreference Model

Features for resolving coreferences (some of which are found in Tables 3.3 and 3.6) continue to fade away, as newer research rely exclusively on neural architectures to detect the intrinsic features of text. An example of this is the SpanBERT model (Joshi et al., 2019a), which is completely isolated from linguistic features or entity information when masking and predicting spans. The model performs better when randomly sampling spans, as discussed in an episode of the Data Skeptic podcast[5] with Omer Levy (one of six authors). As much as it is interesting to understand specifically *why* the performance improves when altering it, for the purpose of this thesis, a good model has been defined as one that performs well across several domains of evaluations. Whether this is to be the most complex model, or a rule-based one will be uncovered throughout the experimentation chapters.

[4] https://github.com/ShannonAI/CorefQA/issues/15
[5] https://podtail.com/en/podcast/data-skeptic/spanbert/ timestamp: 16m20s-17m

4. Data

This chapter covers the data to be used for the experiments throughout the rest of the thesis. Existing datasets for Coreference Resolution (CR) and Entity-level Sentiment Analysis (ESA) are covered Section 4.1 and 4.2. An analysis of the data is found in Section 4.3 and the chosen datasets are listed in Section 4.4. Finally, addressing the issues on existing data for ESA, a dataset has been created using the Strise Knowledge Graph and annotated by Distant Supervision (DS). This process is thoroughly documented in Section 4.5.

4.1. Datasets for Coreference Resolution

As the desired data genre for consideration in this thesis primarily regards news, available datasets are thus divided into in-domain (including news-like text) and out-of-domain (all other sources). Findings in Zeldes (2017) show that different genres or topics may correlate with different patterns of coreference types. By separating genres in a model, testing on domain-specific data, this may be confirmed.

4.1.1. In-domain

In-domain datasets are defined as those that have some correlation with news articles and other online documents, filings, and so on. Only two discovered datasets contain a subset of news-related data: OntoNotes and GUM.

OntoNotes

The CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes (Pradhan et al., 2012) provided a large corpus, labeled in three languages: English, Chinese and Arabic. The dataset contains masked mentions that must be predicted based on their related entity. The dataset is based around telephone conversations, newswire, newsgroups, broadcast news, broadcast conversation, weblogs. By extracting the data related to newswire, newsgroups and broadcast news, this is considered in-domain. This dataset kick-started the field of modern CR, and works as the basis for nearly all models since its conception – thus it is necessary to include for evaluation. The format used, as adopted from Pradhan et al. (2011), is coined *CoNLL*. Ratio of in-domain to total items: $922/3493 \approx 26\%$.

GUM

The Georgetown University Multilayer Corpus (GUM) contains English coreference annotated texts from several sources. Topics covered are **interviews**, **news**, travel guides, how-to guides, **academic writing**, **biographies**, fiction and online forum discussions. Relevant topics marked in boldface. This is by far the most recent dataset, which launched its 6.0 version in March 2020[1]. The corpus is, unfortunately, quite small. Its current version contains 130 documents in total, 21 of those news-related. In spite of this, its well-documented approach and annotation quality makes it a valuable addition for evaluation datasets. Its format is similar to that of CoNLL, but requires extensive parsing to be converted for comparisons. Ratio of in-domain to total items: $21/130 \approx 16\%$.

4.1.2. Out-of-domain

These are datasets that do not necessarily have any relation to news, media and other online official documents. However, using out-of-domain data to evaluate may produce valuable information on the model's robustness and performance.

ARRAU

The Anaphora Resolution and Underspecification corpus (Poesio and Artstein, 2008), further explained in Poesio et al. (2018), does not necessarily provide complete coreference labeling, but as seen in Figure 2.1, anaphora resolution is part of coreference – thus a possible addition to a system focusing on the different aspects of coreference.

Character Identification

Chen and Choi (2016) published a character identification dataset based on the TV show *Friends*, with annotated transcriptions relating to characters in the show. While speech is outside the scope of this thesis, it could be used for other applications of coreference – if not to only evaluate their generalizability.

GAP Coreference Dataset

GAP, Gender Ambiguous Pronouns, is a gender-balanced dataset by Webster et al. (2018) with pairs of ambiguous pronouns and antecedent names. This dataset was involved in a shared task[2] and several great models were developed for the resolution of pronouns and antecedents. The dataset and task of the GAP dataset, however, rely on already masked references, not the detection of coreference scopes. Thus, it is not included for the main experiments of this thesis.

[1] `https://github.com/amir-zeldes/gum/releases`
[2] `https://www.kaggle.com/c/gendered-pronoun-resolution`

LitBank

LitBank is a relatively small, but detailed corpus of 100 literary works in the fiction genre, provided by Bamman et al. (2020). The coreferences are related to the named entity categories of people, facilities, locations, geo-political entities, organizations and vehicles. All named entities are of relevance for the ultimate goal of applying coreference to the real-world news domain – which makes it valuable to achieve notable evaluation scores for this dataset. The format follows the CoNLL standard with slight modifications – requiring manual intervention.

ParCor

ParCor (A Parallel Pronoun-Coreference Corpus) is a parallel English-German pronoun-coreference corpus based on transcriptions from TED Talks and EU Bookshop documents (Guillou et al., 2014). While the labeled parallel texts are irrelevant for this task, it may see further use in multilingual approaches for coreference.

PreCo

PreCo (Preschool Vocabulary for Coreference Resolution) Chen et al. (2018), is a rather large dataset, comprising 38,000 documents and 12.4 million words – 10 times larger than OntoNotes. The annotated texts use simple vocabulary, matching that of English-speaking preschoolers. The authors present experiments allowing for more efficient error analysis than that of OntoNotes. Its format is slightly different than the standardized CoNLL-format, which requires it to be manually translated in order to perform evaluations. However, the amount of data makes this desirable to use for experiments.

WikiCoref

WikiCoref (Ghaddar and Langlais, 2016) is a corpus with annotated coreference on English Wikipedia articles. With a tiny amount of annotated documents ($n = 30$), it may be used to further build upon out-of-domain evaluation data. Its format is unlike any previously covered dataset (XML), and its relatively small size makes it tough to justify converting it to a supported format. Additionally, this dataset has been evaluated in great detail by Moosavi (2020) on several models mentioned throughout related work (Section 3.2).

4.2. Datasets for Entity-level Sentiment

Throughout the literature review in the specialization project (Jørgensen, 2019), several datasets were discovered. Although Sentiment Analysis (SA) is a popular research topic in NLP, datasets for ESA are scarce (Sukthanker et al., 2018). In fact, there are currently no available datasets for the desired task of this thesis, wherein targets of the expressions are isolated entities, residing in longer documents. Existing datasets tend to focus on

the *aspects* of entities, being implicit properties of an entity (such as "screen" for the entity "phone"), as well as annotating targets mostly on a sentence-level basis, not on paragraphs or longer texts. Performing analysis of the datasets (Section 4.3.3), it can be observed that this data is not well suited for coreference augmentation, and motivates the need for a new dataset for this task.

4.2.1. SemEval

SemEval, the International Workshop on Semantic Evaluation, has published open competitions – called *shared tasks* – since 1998, in a diverse set of Natural Language Processing (NLP) topics. A total of three shared tasks from 2014 and 2017 were considered, where certain *subtasks* related to the classification of targeted sentiment can be found.

SemEval 2014 – Task 4

The subtask of *Aspect Term Polarity* by Pontiki et al. (2014) aims to predict the sentiment polarity of a specific aspect in a set of reviews for Laptops and Restaurants (550 total documents), obtained from Yelp and Amazon. The Laptop and Restaurant reviews are separate datasets.

SemEval 2017 – Task 4

Rosenthal et al. (2017) released three datasets of varying topics of Twitter data, intended to be used for the subtask *Sentiment Analysis in Twitter*. One of the three datasets are labeled with negative/neutral/positive sentiment polarities, the other two with binary positive/negative. Thus, only the former is selected for possible evaluation.

SemEval 2017 – Task 5

For the subtask *Fine-Grained Sentiment Analysis on Financial Microblogs and News*, Cortis et al. (2017) provided a dataset of annotated news headlines. In the dataset, stock symbols and company names are labeled with sentiment scores in the range $[-1, 1]$ – which must be translated into other datasets' ternary labeling of $-1, 0, 1$, or kept as-is for fine-grained annotation tasks. Data sources include Yahoo Finance and other websites.

4.2.2. ACL-14

Dong et al. (2014) released a dataset for target-dependent classification of Twitter data. The annotation scheme here has been adopted for all evaluations of sentiment further in this thesis, minimizing extraneous notation. The scheme is line-based, in batches of three: **text, target, sentiment**, where the sentiment is labeled negative, neutral and positive as -1, 0, 1, respectively. The scheme is illustrated on the next page. Note that T indicates a masked target to predict.

```
did SomeCompany steal the code for a $T$ application?
ArbitraryOperatingSystem
0
stoked on the new $T$ game!
ArbitraryGameDeveloper
1
...
```

4.2.3. SentiHood

Saeidi et al. (2016) released a dataset containing conversations on urban neighborhoods from online forums. The annotation style closely matches the desired input data from real-world applications, illustrated in Table 4.1. The labels are restricted to binary sentiment, and the data is consequently not of interest for this thesis. However, this type of data may be relevant for more fine-grained aspect applications.

Sentence	**Labels**
Hampstead area, more expensive but a	(Hampstead, price, Negative)
better quality of living than in **Tufnell Park**	(Hampstead, live, Positive)

Table 4.1.: SentiHood annotation scheme

4.3. Dataset Inspection and Analysis

By inspecting the data to be used for further modeling, a better understanding of the models' faults and feats may be obtained. For the CR datasets, four were deemed necessary to include (as determined throughout Section 4.1) before proceeding with the analysis. The four datasets had to be manually processed and translated into a unified format beforehand. For Entity-Level Sentiment, all listed datasets were included to perform a more in-depth analysis, as sentiment analysis datasets require minimal overhead to include (in contrast to the CR data). This analysis was also motivated by the suggestions of Sukthanker et al. (2018) for more exhaustive evaluations of SA datasets – before continuing with application of CR.

4.3.1. Unification of Coreference Data

For a better overview of the most promising datasets (*OntoNotes*, *GUM*, *LitBank* and *PreCo*), an analysis of their content is presented. Firstmost, the four datasets all follow a different annotation scheme, illustrated in Table 4.3, which has to be converted into a unified format. The reason for previous lack of cross-domain evaluation might be due to this specific task of unification being a necessary first step – which is also supported by the claims on non-CoNLL corpora by Moosavi (2020). The recency and in-depth examination of coreference found in Moosavi (2020) indicates that there are, in fact, no

Source	Format	File size (MB)	Removal actions
Pradhan et al. (2012)	.parse, .prop, .sense, .coref, .names, .lemma	803/55/100 958 total	-
Pradhan et al. (2012)	.conll	188/24/25 237 total	Merging of multiple files into .conll
Lee et al. (2018)	.jsonline	45.4/5.9/5.6 56.9 total	POS-tags, lemmas, word sense
This work	.coreflite	12.3/1.6/1.5 15.4 total	Speaker info, constituents, entity metadata

Table 4.2.: OntoNotes 5.0 dataset processing steps. File size is separated into train/dev/test and total size

Dataset	File format	Coreference format
Minimized OntoNotes	jsonline	$[Ms_{token}, Me_{token}]$
GUM	conll	chain of $\{entitytype + iterated\ index\}$
LitBank	conll	chain of $\{iterated\ index\}$
PreCo	jsonline	$[sentence\ index, Ms_{subtoken}, Me_{subtoken}]$

Table 4.3.: The four used datasets for CR and their file format as well as coreference annotation format. Me/Me denotes a mention with its start and end indices. *subtoken* denotes a sentence-level (local) token, where *token* is a global token.

current cross-corpora evaluations of extent in any other literature. While the datasets have similarities (e.g. the CoNLL format), minor intricacies cause incompatibility when parsing. To combat this, a new, simple, unified format is defined, based around the minimization process of the OntoNotes dataset (Pradhan et al., 2012) by Lee et al. (2018)[3]. The format is coined "CorefLite" (a lightweight coreference format). Table 4.2 illustrates the transition of the OntoNotes dataset and its file size from its original format to CorefLite. The same process for the remaining datasets can be found in Table 4.4. Note that the reduced file size is not the goal for the format, but rather to reduce all four datasets into the same format, with the exact same input fields (tokens and clusters). The extent of the reduction is rather an indication of how much extraneous data is contained within these datasets. The CorefLite structure is shown below, where the specific coreference clusters format is described in Background, p. 25.

```
{
    doc_key:     # document identifier,
    tokens:      # list of all tokens,
    clusters:    # coreference clusters
}, { ... }
```

[3] https://github.com/kentonl/e2e-coref/blob/master/minimize.py

Dataset	Initial size (MB)	Coreflite size (MB)
GUM	1.7	1.3
LitBank	12.6	2.0
PreCo	154.2	146.5

Table 4.4.: The remaining datasets and their parsed file size

Dataset	N	L_{avg}	L_{min}	L_{max}	C_{total}	C_{avg}/doc
OntoNotes (dev)	343	475.52	33	2314	4545	13.25
GUM	130	872.11	165	1866	4401	33.85
LitBank	100	2105.32	1999	3419	4975	49.75
PreCo (dev)	500	332.38	50	966	31793	63.59

Table 4.5.: Overview of coreference dataset features, sampled on subsets with similar sizes. N denotes the total number of documents, L denotes the document length, C denotes coreference links.

Dataset	N	L_{avg}	L_{min}	L_{max}	C_{total}	C_{avg}/doc
SemEval 2014 – Task 4	7694	93.68	8	470	2207	0.29
SemEval 2017 – Task 4	2872	104.13	26	144	1060	0.37
SemEval 2017 – Task 5	1156	56.54	25	112	48	0.04
ACL-14	6940	89.17	10	161	2214	0.32
SentiHood	4333	77.48	7	564	658	0.15

Table 4.6.: Entity-level sentiment dataset features. N denotes the total number of documents, L denotes the document length, C denotes coreference links generated with NeuralCoref.

4.3.2. Coreference Dataset Analysis

The analysis process, dependent on the creation of the unified format, involve fairly simple categorization of the data involved in each dataset, to use as a baseline for future datasets. In Table 4.5 an overview of the four chosen datasets can be found. For *OntoNotes* and *PreCo*, smaller subsets of the data was used, with similar file sizes of the smaller datasets *GUM* and *LitBank*. Observe the number of C_{total} and C_{avg}/doc, as these will be calculated for the same analysis of entity-level sentiment data in next section. Further, the *PreCo* dataset has a much higher occurrence of coreference links with a lower average document length. Handling this type of data might be troublesome for models trained strictly on the data in *OntoNotes*. A visualization of the relationship between coreference links and document length can be found in Appendix C. Further analysis will occur throughout the Preliminary Experiments chapter on CR (Chapter 7).

4.3.3. Restrictions of Entity-Level Sentiment Data

The document length is crucial in order to resolve coreferences. Thus, the document length of each dataset is compared, together with the number of detected coreference links. Detailed plots from this analysis can be found in Appendix B. Presented in Table 4.6 is an overview of the analysis. For the datasets *SemEval 2014 – Task 4*, *SemEval2017 – Task 4* and *ACL-14* there are approximately one coreference cluster in every three documents, using the **NeuralCoref** CR model. Further, the sentiment polarity distribution was looked at, illustrated in Figure 4.1. None of the available datasets were balanced (i.e. having an approximately even distribution between its classes), which might result in issues for evaluating the news domain. There are currently no available datasets that show distributions on sentiment for full news texts, so there is currently no way of verifying whether this data is representative across domains. Continuing based on this analysis, the datasets *ACL-14* and *SemEval 2014 – Task 4* provide the most distributed sentiment, as well as a fair amount of coreference links.

4.4. Selected Datasets

The final datasets used for both topics are listed in Tables 4.7 and 4.8.

Dataset	**Relevant topics**	**Size (#documents)**
OntoNotes	Partly	3,493
GUM	Partly	130
LitBank	No	100
PreCo	Unknown	38,000

Table 4.7.: Selected datasets for Coreference Resolution

Dataset	**Data type**	**Size (#documents)**
ACL-14	Twitter data	6,940
SemEval 2014 – Task 4	Online reviews	7,694

Table 4.8.: Selected datasets for Entity-Level Sentiment Analysis

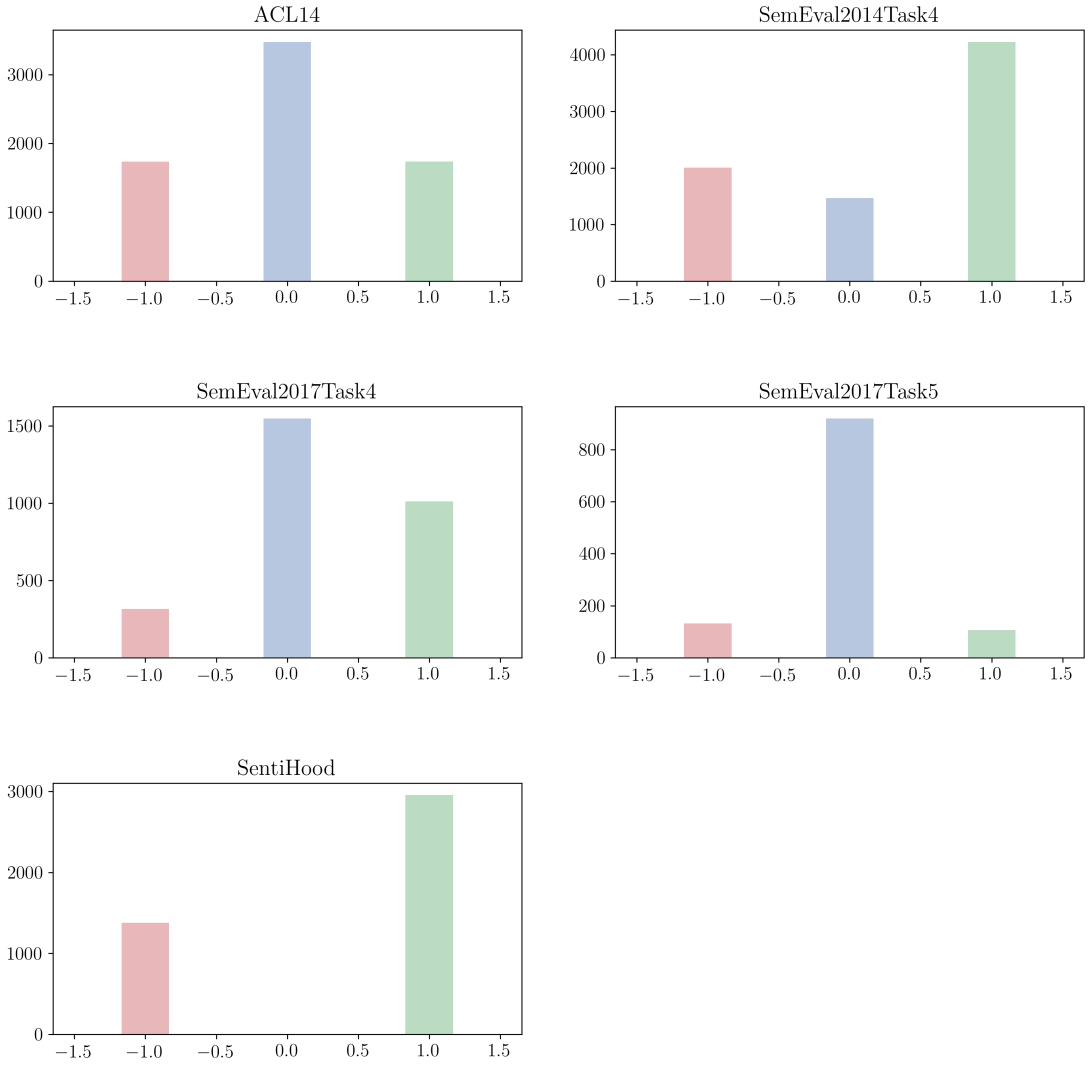


Figure 4.1.: Sentiment polarity distribution for five datasets, with the labels −1 (negative), 0 (neutral) and 1 (positive).

4.5. Dataset Creation with Distant Supervision and World Knowledge

The discussed issues with available datasets for ESA are too great to ignore. To address these, a dataset has been created using the knowledge graph at Strise, to jointly access news texts, official documents and more, while adding to this data with entity information and world knowledge. The process is largely based around a 4-step pipeline, described in detail in the coming sections.

1. Gathering data
2. Parsing data
3. Distant supervision labeling
4. Data analysis and verification

4.5.1. Gathering Data

At the very start of the pipeline comes the decision on what metadata to use. This includes topics from WikiData (which are interconnected to the Strise Knowledge Graph) as well as what timeframe to gather data from. An event is the selected term to refer to any news article or document obtained from the Strise Knowledge Graph. The selected topics are rather arbitrary, and focused on grouping distinguishable topics that may be evaluated separately. The time frame (2018–2020) was selected to gather recent data.

1. Iterate the four defined topics: (tech [technology/science], business [finance/economics], politics and sports)
2. Iterate years (2018, 2019, 2020)
3. Iterate quarters (Q1, Q2, Q3, Q4)
4. Gather the same amount of events (400) for each quarter

This process resulted in 1600 events per year, 3600 in total (one quarter for 2020). Two final datasets were created. One in which the data from 2020 was set aside as evaluation data (400 events), and the other considered all data, to be split programmatically (documented in the Experimentation chapter).

4.5.2. Parsing Data

The parsing step accesses the gathered data, structuring it in a machine readable format (CSV). In the process, entities are filtered on their inherent properties, based on its features found in the knowledge graph. The knowledge graph contains intricate details, to low levels such as the mass and CPU of a smartphone – hence the need for filtering. The filtering process starts by addressing the relations of entities, as seen in Table 4.9.

These can be understood by asking the question "Can the relation for an entity X be used to refer to X in the scope of an event?". Although vague, the goal is to generate possible candidates for referring to a given entity – treating them as one and thus more data for the entity.

Disambiguation and Pseudo-References The set of relations is the result of massive trial and error. Omitting filtering completely yields entities such as "4, stop, point, week, month", which are deemed irrelevant. This process is highly empirical and revolves around inspecting the data for an acceptable outcome, making sure to include wanted entities and simultaneously discern between ambiguities (e.g. "apple" [fruit] and "Apple" [company]). Although the process requires manual, meticulous work, it allows for extremely customized data retrieval, which may apply well for other tasks. For instance, one may only obtain information on the CEOs of companies, and thus making way for analyses like "How are CEOs represented in the news?" and so forth – the possibilities here are endless! Moving on, to illustrate the reconciliation process, a text is given as an example (from *sciencebusiness*[4]), where the entity *NTNU* is in focus. The *NTNU* entity is linked to WikiData ID *Q314536*, which enables the system to provide references from the knowledge graph (as seen in Table 4.10). Reconciling the entity with the obtained references, all references are updated as the entity itself (generating pseudo-references), finally marked as **T** (target).

> NTNU researchers recently figured out a whole new method for testing people for the coronavirus. The university is now producing tests on a continuous basis, under the auspices of the Norwegian Directorate of Health.

Parsed:

> **T** researchers recently figured out a whole new method for testing people for the coronavirus. **T** is now producing tests on a continuous basis, under the auspices of the Norwegian Directorate of Health.

Error Handling Incorrect masking may occur when handling certain prefixes, suffixes and plural forms (e.g. "T-esque" and "T's"). This was handled by applying post-processing techniques to the tokens, merging them as one: *Google-esque* $\longrightarrow$ *T-esque* $\longrightarrow$ *T*. Another issue was inter-entity masking (e.g. regarding *Trondheim* as *NTNU*). A workaround was established by doing an initial pass through all data and obtaining a list of unique entities – a lookup-dictionary. This way, the relation *Trondheim* will not be attributed to *NTNU* if *Trondheim* exists in the lookup-dictionary.

[4] `https://sciencebusiness.net/network-updates/ntnu-establishes-factory-produce-coronavirus-tests`

Defined Category	Accepted Relations
person_relations	[occupation, profession, position_held, spouse]
business_relations	[topic_itpc, owner_of, leader, chief_executive_officer, headquarters_location, parent_organization, industry, investor]
location_relations	[capital, country, continent, executive_body, head_of_government, head_of_state, location, located_in_the_administrative_territorial_entity]
product_relations	[inception, manufacturer, developer, discoverer_or_inventor, designed_by, founded_by]
arts_relations	[composer, director, distributor, genre, producer, production_company]
other_relations	[creator, owned_by, used_by, parent_concept, sport]

Table 4.9.: Entity relations for filtering

Field	Data
ID	Q314536
Name	NTNU (Norges teknisk-naturvitenskapelige universitet)
Description	University in Trondheim, Norway
Aliases	- Norges teknisk-naturvitenskapelige universitet - NTNU (Norges teknisk-naturvitenskapelige universitet) - Norges Teknisk-Naturvitenskapelige Universitet - NTNU - Norwegian University of Science and Technology
Member of	- Top International Managers in Engineering - European University Association
Country	Norway
County	Trøndelag
Rector	Gunnar Bovim
Instance of	University
Located in	Trondheim

Table 4.10.: Entity information on NTNU from the Strise Knowledge Graph. Note that Gunnar Bovim is currently not the Rector. The information is incorrect – but only for a given time frame. This sheds light on the importance of the time aspect for these tasks. If the related article is dated before August 2019, it is certainly the correct information.

Storing Entity Segments Using the masked entity data, segments were split and stored (along with the necessary entity information) in a CSV format with the following structure: *{text_id, entity_id, name, description, segment, references}*. The data at this step includes segments with masked entities, in addition to permutations occurring where another reference to the entity is found within the same segment – producing a new masked object.

4.5.3. Distant Supervision Labeling

All segments with their masked entities were passed through a 2-step sentiment model. The models consist of Vader (Hutto and Gilbert, 2014), a lexicon and rule-based classifier, and DistilBERT (Sanh et al., 2019), a BERT-based model pre-trained on binary sentiment data. The labeling process is defined as follows, simulating a ternary classifier:

1. Label neutral texts by verifying the outcome of Vader. Return neutral and its respective score (ranging $[0-1]$).
2. If not neutral, classify with DistilBERT. Return positive/negative and its respective score.

Vader – Empirical Definition of Neutrality

Vader provides two scores for verifying neutral texts: polarity score and compound score. The compound score is crucial for determining neutrality, as it sums the valence scores, that is the *affective quality* of a word – good/bad. A document may commonly be misclassified as neutral if the negative and positive words cancel out, but its valence score would thus be high. As suggested by Hutto and Gilbert (2014), the threshold was evaluated and set to 0.12 (diverging from the default 0.05) as, during manual inspection, news-related data frequently include high-valence words, even if the conveyed sentiment is neutral.

DistilBERT SST-2 – Efficient Binary Classification

DistilBERT (Sanh et al., 2019) is a lightweight BERT-based model, pre-trained on SST-2 – the Stanford Sentiment Treebank (Socher et al., 2013) – which is how distant supervision comes into play. The model retains 97% of the original BERT performance, while its size is reduced by 40%. Its accuracy on the SST-2 task is 92.7%, with the current state-of-the-art T5-11B model with 97.1% (Raffel et al., 2019). The latter model is much, much larger (116× the size), and would severely impact the efficiency of the labeling process. Another light model is ALBERT (Lan et al., 2019). However, the ALBERT model of similar size to DistilBERT (60M vs 66M parameters) reports an accuracy of 92.4%. The largest ALBERT model (xxlarge, 235M parameters) shows a score of 95.2%, but again, is much less efficient than DistilBERT. The fine-tuning configuration of DistilBERT can be seen in Appendix H. Additionally, the DistilBERT model is available through a high-level API through the Transformers library (Wolf et al., 2019), allowing for convenient integration into systems.

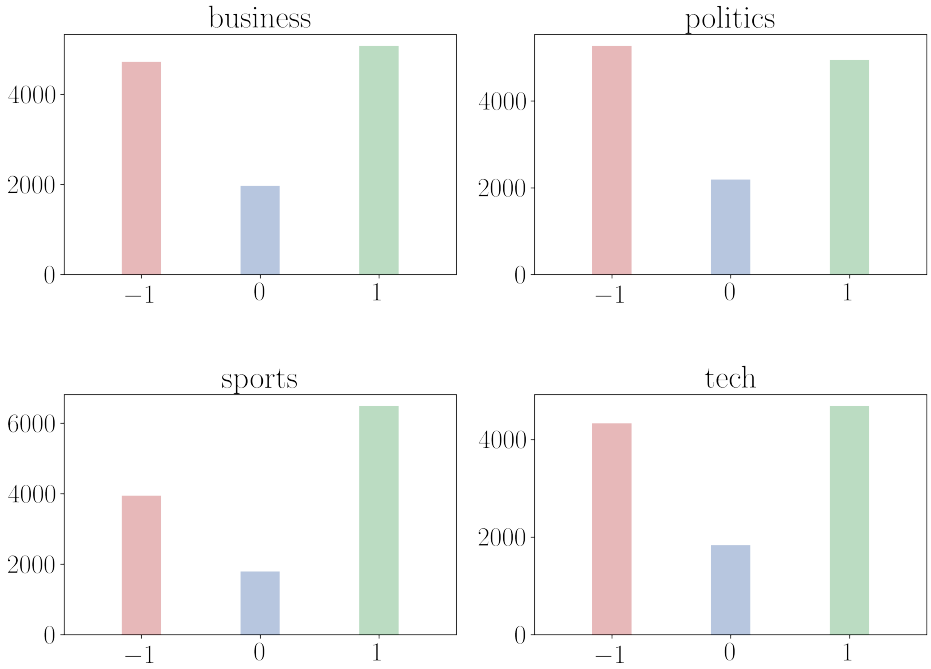


Figure 4.2.: Sentiment polarity distribution by topic in the generated dataset, with the labels −1 (negative), 0 (neutral) and 1 (positive).

4.5.4. Data Analysis and Verification

As briefly noted earlier (Section 4.3.3), it is desirable to obtain a balanced dataset with respect to the outcome of the labeling process. From the automatic labeling, after long-going manual tuning, the dataset upheld a fair balance between positive and negative – with fewer neutral classifications. Empirically evaluating the extremes of each label was necessary to detect unwanted entities and wrongly classified documents. After evaluation, if the results were unsatisfactory, the process had to start over with the parsing of data. A summary of the final sentiment distribution can be found in Figure 4.2 and 4.3.

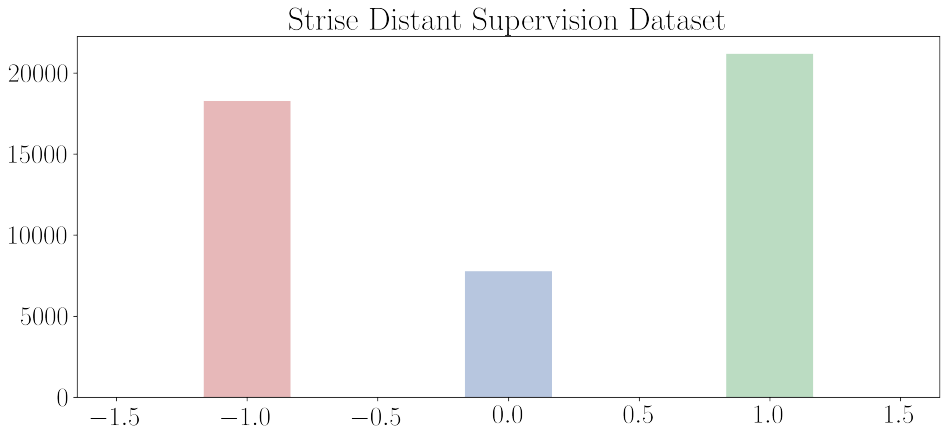


Figure 4.3.: Sentiment polarity distribution for the entire generated dataset (combined topics).

5. Architecture

This chapter presents architectures for several systems used in order to run experiments (chapters 6 and 7). Firstly, to provide an overview of the chapter, the connections between the systems developed throughout the thesis are illustrated. Following this, the architecture regarding data handling and format unification of existing datasets is described. Continuing, architectures implementing existing Entity-level Sentiment Analysis (ESA) and Coreference Resolution (CR) models for evaluation are documented, using the output of the aforementioned unification process. Lastly, the chapter is concluded with the architecture for the generated dataset.

5.1. An Overview

To better visualize how the forthcoming systems and architectures are connected, an architectural description can be found in Figure 5.1. To clarify further illustrations, the topics CR, ESA and systems regarding Strise, will follow the color schemes in the figure – blue, red and yellow respectively. White indicates processing not conforming to a specific topic. Anything relating to the final model will be marked green.

5.2. CL-Eval – Evaluation Framework for Coreference Resolution

This section is dedicated to the developed evaluation framework, used to evaluate selected in- and out-of-domain datasets for CR. The name CL-Eval is shorthand for *CorefLite-Evaluation*, based on the defined unified format for CR datasets. Its architecture is shown in Figure 5.2.

5.2.1. CorefLite – a Unified Format for Coreference Resolution

As presented in the Data Section (4.3.1), the CorefLite (or .coreflite) format was defined to unify selected datasets as one format, enabling ease of evaluation and analysis. The architecture is mainly based around the parsing of datasets to extract data in a desired format, described in 4.3.1, and shown below for simplicity:

```
{ doc_key:     # document identifier,
  tokens:      # list of all tokenized words,
  clusters:    # coreference clusters
}, { ... }
```

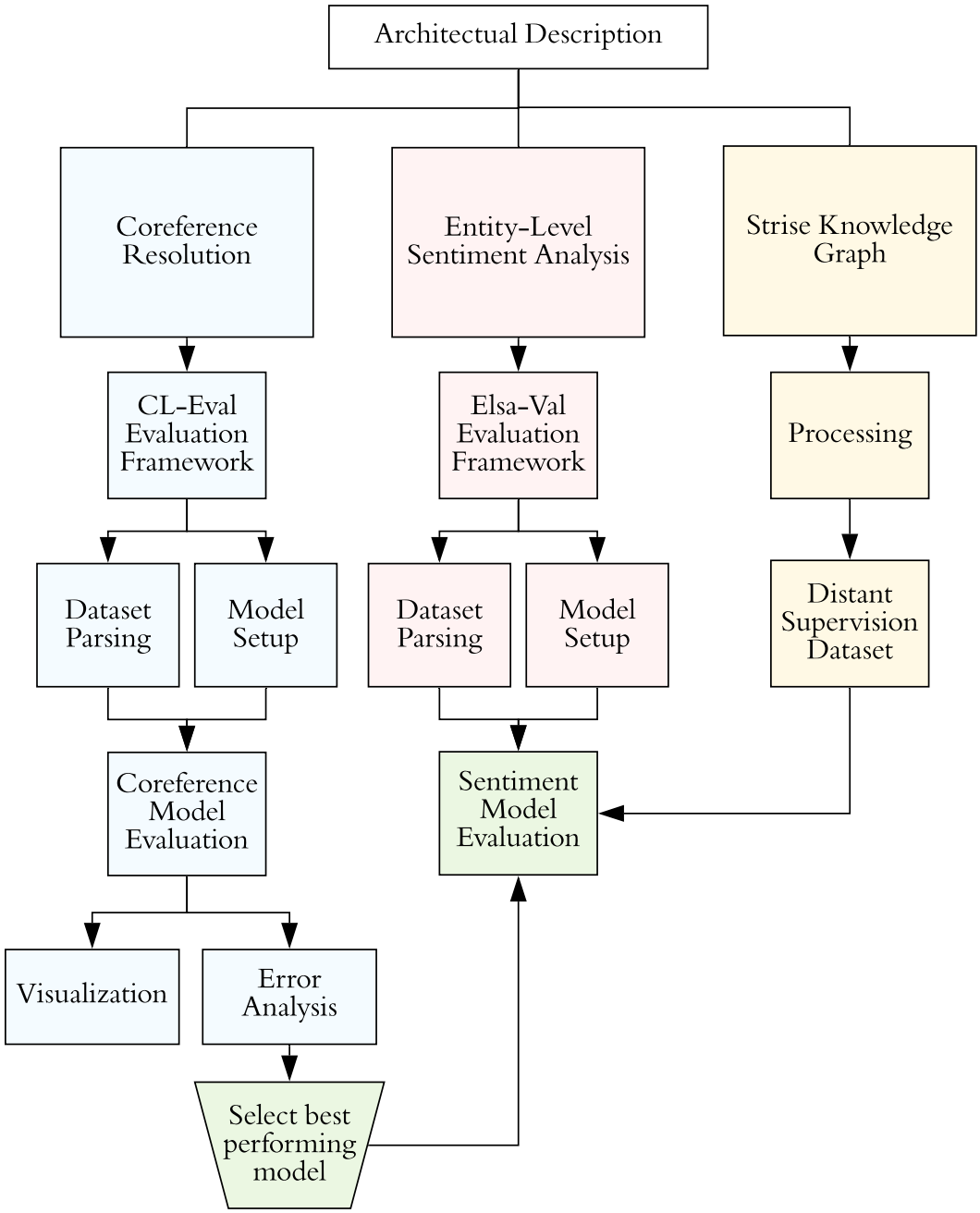


Figure 5.1.: Architectural description

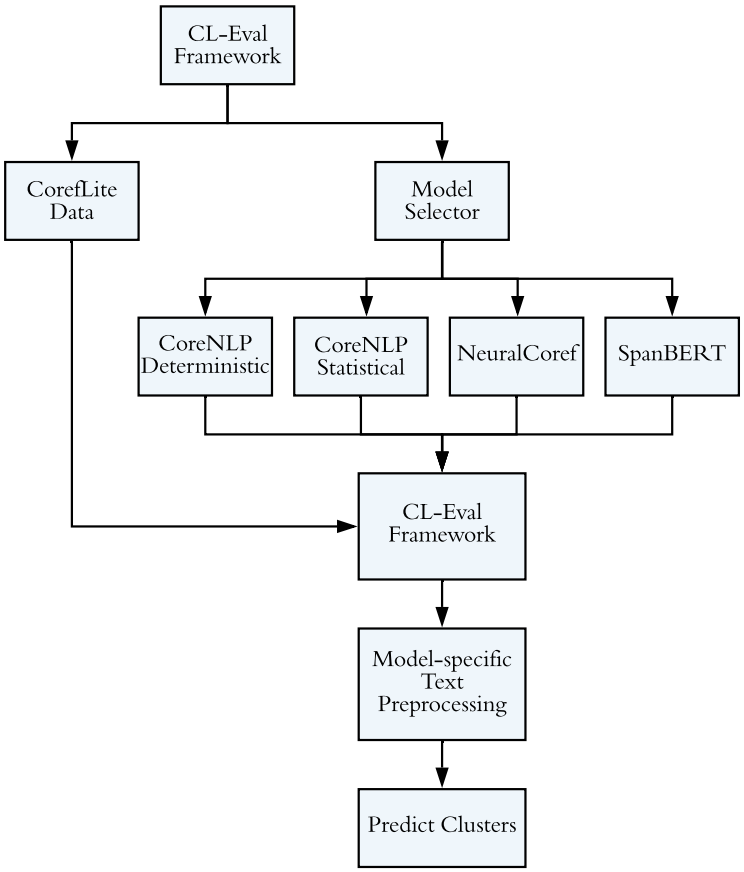


Figure 5.2.: CL-Eval Framework Architecture

The four datasets required different parsing in order to unify the formats. The process for each dataset is briefly described below:

OntoNotes

As discussed throughout Chapter 4, the OntoNotes has already seen major parsing through the minimization process of Lee et al. (2018). Conversion of the dataset was done in three steps:

1. Flatten sentences to obtain the segmented tokens
2. Extract coreference clusters
3. Correct unwanted tokens that are badly handled by most models

PreCo

PreCo defines its coreference clusters in triples (sentence index, mention start, mention end). This had to be converted to a tuple of (mention start, mention end), thus the following steps were required:

1. Update respective mentions with the number of previous tokens seen throughout the text, at a specific sentence index
2. Correct mention indices after removing newlines and other invalid tokens

GUM

Both the GUM and LitBank datasets had separate files for each document, in CoNLL format. The CoNLL format was handled as lists, where the coreference index had to be manually parsed. To illustrate the annotation scheme, observe the following snippet[1]:

```
12 University (organization-7
13 of _
14 Portsmouth (place-8
15 ,_
16 United (place-9
17 Kingdom organization-7)place-8)place-9)
18 Andrew (person-10
19 Beresford person-10)
..
155 UK (place-9)
```

[1]Full document available at https://github.com/amir-zeldes/gum/blob/master/coref/conll/GUM_academic_art.conll

The grouped notation on line 17 indicates that the spans "University of Portsmouth, United Kingdom", "Portsmouth, United Kingdom" and "United Kingdom" are mentions. The coreferences (as seen on the last line) are annotated by referring to the antecedent mention label. In order to convert this to the coreflite format, the following steps were required:

1. Reformatting of the coreference texts
2. Extraction of singular token mentions, not split over several lines (e.g. (*person-89*))
3. Handle and merge open-ended references, as can be observed on line 12 – 18 in the example above
4. Convert from a dictionary of *{entity: mention indices}* to a list of *[mention start, mention end]*

Parsing the example, the resulting output would be *[[12, 17]], [[14, 17]], [[16, 17], [155, 155]], [[18, 19]]*. Note that both mentions of "United Kingdom" occur in the same inter-entity cluster.

LitBank

Largely similar to that of GUM, with slight annotation and format changes, as can be seen below[2]:

```
24_o_pioneers_brat 0 28 the _ _ _ _ _ _ _ _ _ (13
24_o_pioneers_brat 0 29 north _ _ _ _ _ _ _ _ _
24_o_pioneers_brat 0 30 end _ _ _ _ _ _ _ _ _
24_o_pioneers_brat 0 31 of _ _ _ _ _ _ _ _ _
24_o_pioneers_brat 0 32 the _ _ _ _ _ _ _ _ _ (1
24_o_pioneers_brat 0 33 town _ _ _ _ _ _ _ _ _ 1)|13)
24_o_pioneers_brat 0 34 to _ _ _ _ _ _ _ _ _
24_o_pioneers_brat 0 35 the _ _ _ _ _ _ _ _ _ (14
24_o_pioneers_brat 0 36 lumber _ _ _ _ _ _ _ _ _
24_o_pioneers_brat 0 37 yard _ _ _ _ _ _ _ _ _ 14)
```

The parsing approach was nearly identical to GUM, aside from data pre-processing.

5.2.2. Batch Prediction and Evaluation

As all datasets are now in an identical format, they can be processed through the different models set up. An evaluator has been developed for this purpose, processing data and computing the predicted clusters, evaluating them with the scoring metrics defined in Background (p. 25), MUC, B-CUBED, CEAF and LEA.

[2]Full document available at `https://github.com/dbamman/litbank/blob/master/coref/conll/11_alices_adventures_in_wonderland_brat.conll`

5.2.3. Visualization Module

The visualization module is based on a tiny open-sourced project[3], including a visualization module with wrappers for various coreference outputs. The module was modified to support all models and outputs necessary for experiments to be conducted. It is designed to run within any Jupyter notebook, and its functions are built into the *CL-Eval* framework for quick overview of distinguishable entities and their coreferences. The architecture is built around the *tokens* and *clusters* resulting from a CR model – used to generate unique spans of highlighted text throughout the document.

5.3. Coreference Models

The models set up in the next sections are used to further evaluate existing datasets with the CorefLite format. There are in total four models used. The selection of each model and so forth is described in detail in the Preliminary Experiments (Chapter 6). For completeness, the models are shown in their context to the larger architecture here, and they go by the names: CoreNLP Deterministic, CoreNLP Statistical, NeuralCoref and SpanBERT. The NeuralCoref model had to be worked with in detail to get set up properly, and thus this will be discussed below. The remaining models, CoreNLP Deterministic (Lee et al., 2013) and Statistical (Clark and Manning, 2015), as well as SpanBERT (Joshi et al., 2019a), are all used as described in their respective publications.

NeuralCoref + spaCy

The NeuralCoref model, heavily based around the work by Clark and Manning (2016a), has been set up as two models. The first is built from its source code, trained and evaluated on the OntoNotes dataset. The evaluation scores are not presented by the authors of the model, and will be evaluated in the chapter on Coreference Validation, Section 6.3.2. The other model is a high-level API exposed through local installation of the pre-trained model, integrating with spaCy's pipeline.

[3] https://github.com/sattree/gpr_pub

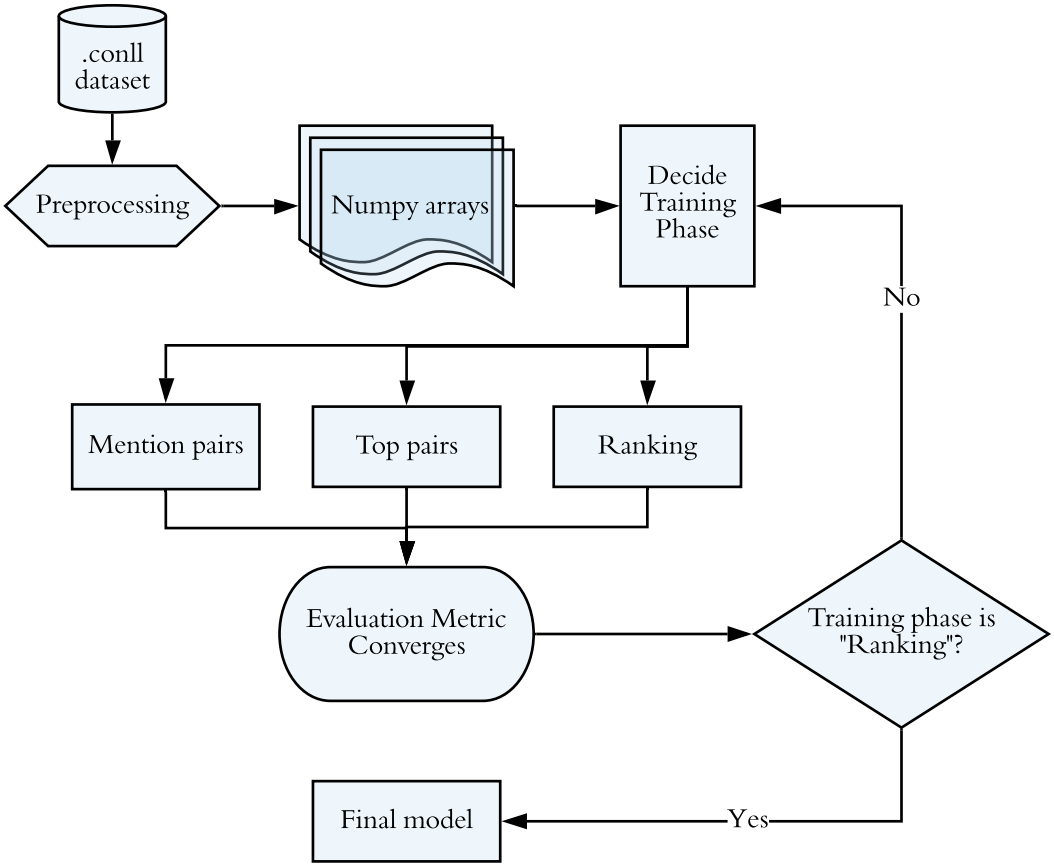


Figure 5.3.: NeuralCoref Training Architecture

First Model – Training Setup

This is the model provided by Hugging Face, with slight modifications to support the IDUN computing cluster. The training of the model runs two feed-forward neural networks, one which outputs the chance of any mention being an antecedent of another mention, the other outputs the change of a mention having *no* antecedent. The network is passed through three steps, as seen in Figure 5.3.

1. Mention pair loss At first, mention pair loss (ℓ) is calculated using a Sigmoid layer (σ) followed by a binary cross entropy loss. N denotes the batch size, w are the weights.

$$\ell(x, y) = -\sum_{n=1}^{N} w_n \left[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n)) \right]$$

2. Top pair loss The second phase calculates the top pairs (both for true and mistaken predictions), by first ensuring all inputs reside in a predictable range, using a *clamped-sigmoid* function (Eq. 5.1) with a defined min and max.

$$\text{clamped-sigmoid}(x) = \begin{cases} min, & \text{if } \sigma(x_i) \leqslant \min \\ \sigma(x_i), & \text{if } \max \leqslant \sigma(x_i) \leqslant \max \\ max, & \text{if } \sigma(x_i) \geqslant \max \end{cases} \tag{5.1}$$

$$s = \text{clamped-sigmoid}(inputs)$$

$$top_{true} = max\Big(log\big(\sigma(s)\big)\Big)$$

$$top_{mistake} = min\Big(log\big(1 - \sigma(s)\big)\Big)$$

3. Ranking loss Finally, a rescaling loss ($\mathcal{R}$) is calculated, following the *Reward Rescaling* approach by Clark and Manning (2016a). $\mathcal{A}$ are all possible antecedents containing mentions, $\mathcal{T}$ is the subset of true antecedents. s are the scores resulted from the *clamped-sigmoid* function (Eq. 5.1).

$$\mathcal{R} = \sum_{i=1}^{N} \max_{a \in \mathcal{A}(m_i)} \Delta(a, m_i) \left(1 + s_m(a, m_i) - \max_{t \in \mathcal{T}(m_i)} s_m(t, m_i)\right) \tag{5.2}$$

To efficiently perform calculations on the vast amount of data to be extracted from coreference data, the model parses input data into a set of Numpy arrays, calculating features, labels, pair lengths and so forth. A full list can be seen in Appendix **??**. This preprocessing step, however, comes at a cost: the (.conll) size of OntoNotes, 237 MB, results in 8600 MB of Numpy array files. This might become an issue for larger datasets. Hyperparameters used are listed in Appendix D.1.

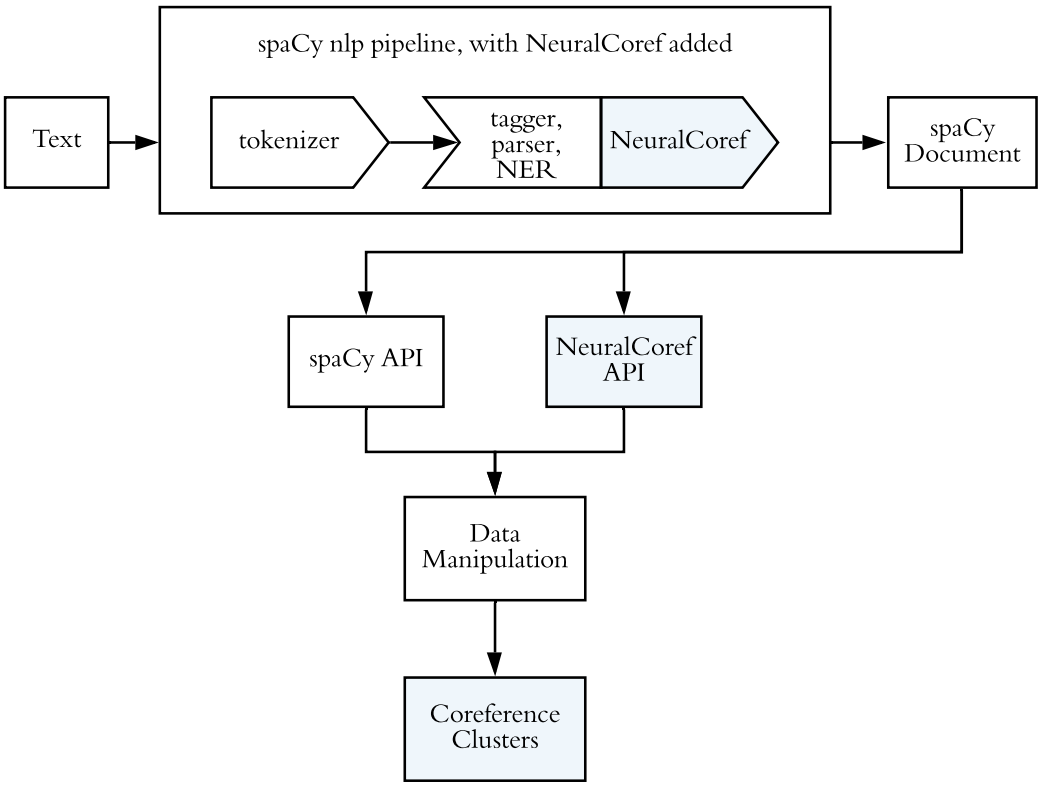


Figure 5.4.: NeuralCoref + spaCy high-level architecture

Second Model – High-level API model with spaCy

Due to the seamless integration with spaCy, the pre-trained NeuralCoref model may be directly added to the spaCy *nlp pipeline* for document handling. This process is illustrated in Figure 5.4. With spaCy's fast dependency parser (Choi et al., 2015), and the API to the pre-trained NeuralCoref model, this architecture is efficient when used to parse large batches of data, and is used for the initial detection of coreference clusters in the the early stages of this thesis. Further documentation on the *nlp pipeline* can be found at spaCy's website[4].

Hyperparameters Various hyperparameters are exposed to the user when integrating NeuralCoref into spaCy. Relevant hyperparameters are described in Table 5.1. See the documentation[5] for more details.

[4] `https://spacy.io/usage/processing-pipelines/`
[5] `https://github.com/huggingface/neuralcoref#parameters`

Parameter	Description
greedyness	Strictness of the algorithm for determining coreference links (the higher value, the more links)
max_dist	Number of mentions to look back for considering antecedents
max_dist_match	Distance between a coreference link to be attributed between two mentions
conv_dict	A dictionary to help the algorithm with resolutions for data it has not yet seen or been sufficiently trained on. Adding *{Covid-19: ["virus", "pandemic"]}* will assist in resolving mentions including "it", "the virus", and so on.

Table 5.1.: A selection of hyperparamters for NeuralCoref

5.4. Elsa-Val – Evaluation Framework for Entity-Level Sentiment Analysis

The evaluation framework developed to set up and evaluate models, coined Elsa-Val, contains separate systems to handle the discovered datasets, as well as training models on this data. The overall architecture can be seen in Figure 5.5. The models are based on the original implementations in a system called *ABSA-PyTorch*[6] by Youwei Song[7]. The system's architecture has been slightly modified to easily evaluate several models in batches, as well as to run on the latest updated technologies. The modified framework is found on GitHub[8].

Model Selection The models selected for evaluations are based on a range of years since development, as well as in increasing complexity. This was done to investigate whether the newer models – performing well on available datasets – would scale consistently with new data, or if an older model architecture could prove its worth. The models implemented are a Long Short-Term Memory (LSTM), originally defined by Hochreiter and Schmidhuber (1997), TD-LSTM – Target-Dependent LSTM (Tang et al., 2016), CABASC – Content Attention Model for Aspect Based Sentiment Analysis (Liu et al., 2018) and LCF BERT – Local Context Focus Mechanisms with BERT (Zeng et al., 2019). The first model, LSTM, was selected as it is considered the foundation for several later models. TD-LSTM was developed to aim the LSTM towards target-dependent classification by implementing a bidirectional LSTM. The CABASC model further attempted to improve these models by incorporating attention mechanisms (Vaswani et al., 2017). The final LCF BERT model was selected as it is currently holding state-of-the-art performance on the chosen datasets. All experiments with these models are found in Chapter 7.

[6] https://github.com/songyouwei/ABSA-PyTorch

[7] Youwei Song is an avid researcher on Entity-level and Targeted Sentiment Analysis, having taken part of multiple systems achieving state-of-the-art performance (Song et al., 2019; Yang et al., 2019a)

[8] https://github.com/ph10m/ElsaVal

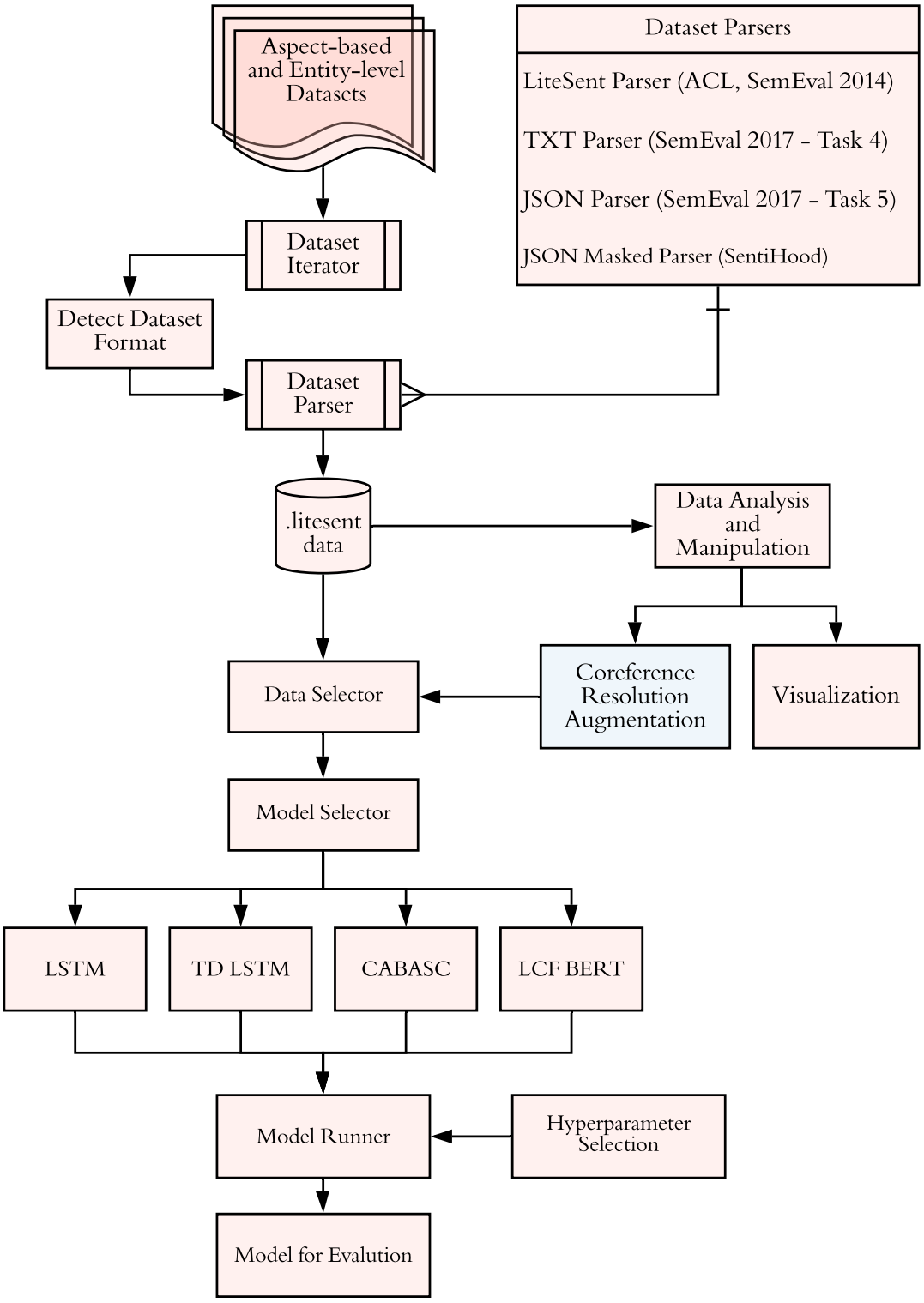


Figure 5.5.: Elsa-Val Architecture.

5.4.1. Annotation Tool

In the very beginning of the thesis, an extremely customizable annotation tool was created, directly interfacing with Pandas Dataframes (see Background, p. 28) – thus the tool is named PandAnnotator. The tool was made to initially label a large quantity of data for ESA, before refocusing the thesis towards CR. The data structure for sentiment was later moved from Dataframes to the .litesent format, allowing manual annotation directly on the data itself. However, there are plenty of usages for annotating directly on a Dataframe – irrespective of the topic – and thus the project has been open-sourced[9]. A screenshot of the Annotation tool is found in Appendix F, illustrating the sentiment towards an entity with a backdrop color (green, yellow and red for positive, neutral and negative), while providing additional information from the Dataframes.

5.4.2. Entity-centric Segmentation Algorithm

A vital part of augmenting the data is done through segmentation of the data based on the outputs from a CR model, as produced by using CL-Eval. The surrounding architecture for the algorithm is visualized in Figure 5.6. There are mainly 6 steps involved to create segments:

1. Generate coreference clusters from a text using a CR model
2. Iterate clusters, look for antecedents matching the target entity in its mentions. If a match is found, mark the given cluster as valid.
3. Iterate all valid cluster mentions, mask it as T and modify the tokens inline (e.g. "the university" to "T")
4. Re-tokenize the final masked tokens as sentences
5. Generate segments by iterating sentences and add each sentence if it contains a masked entity
6. Repeat the previous step until the next mask is within a given distance away from the previous mask (and the segment contains at least *one* mask) until the next sentence contains another mask.

5.5. Generated Dataset

The architecture for the 4-step pipeline described in Section 4.5 can seen in Figure 5.7. The system can be found incorporated into the Entity-level Sentiment Analysis Framework (Elsa-Val) on GitHub[10].

[9] https://github.com/ph10m/PandAnnotator
[10] https://github.com/ph10m/ElsaVal/tree/master/Distant%20Supervision

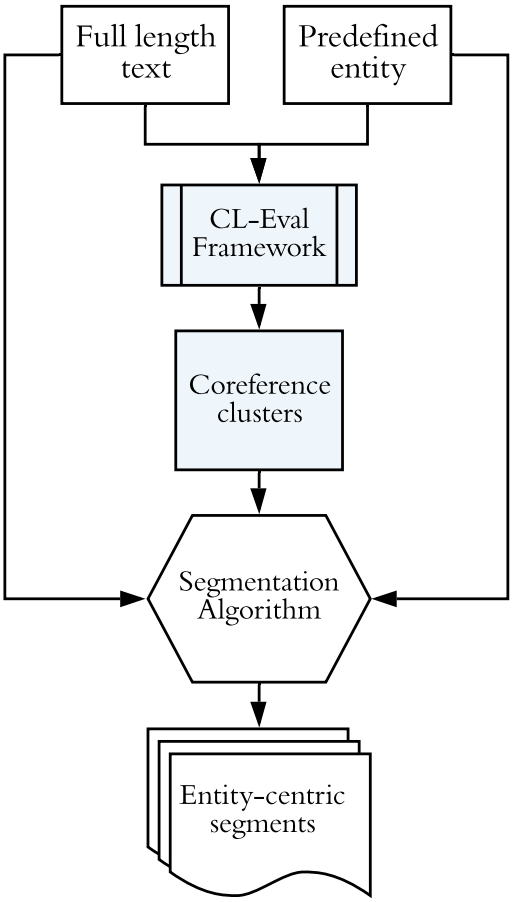


Figure 5.6.: Entity-centric Segmentation Algorithm using the CorefLite Evaluation Framework (CL-Eval).

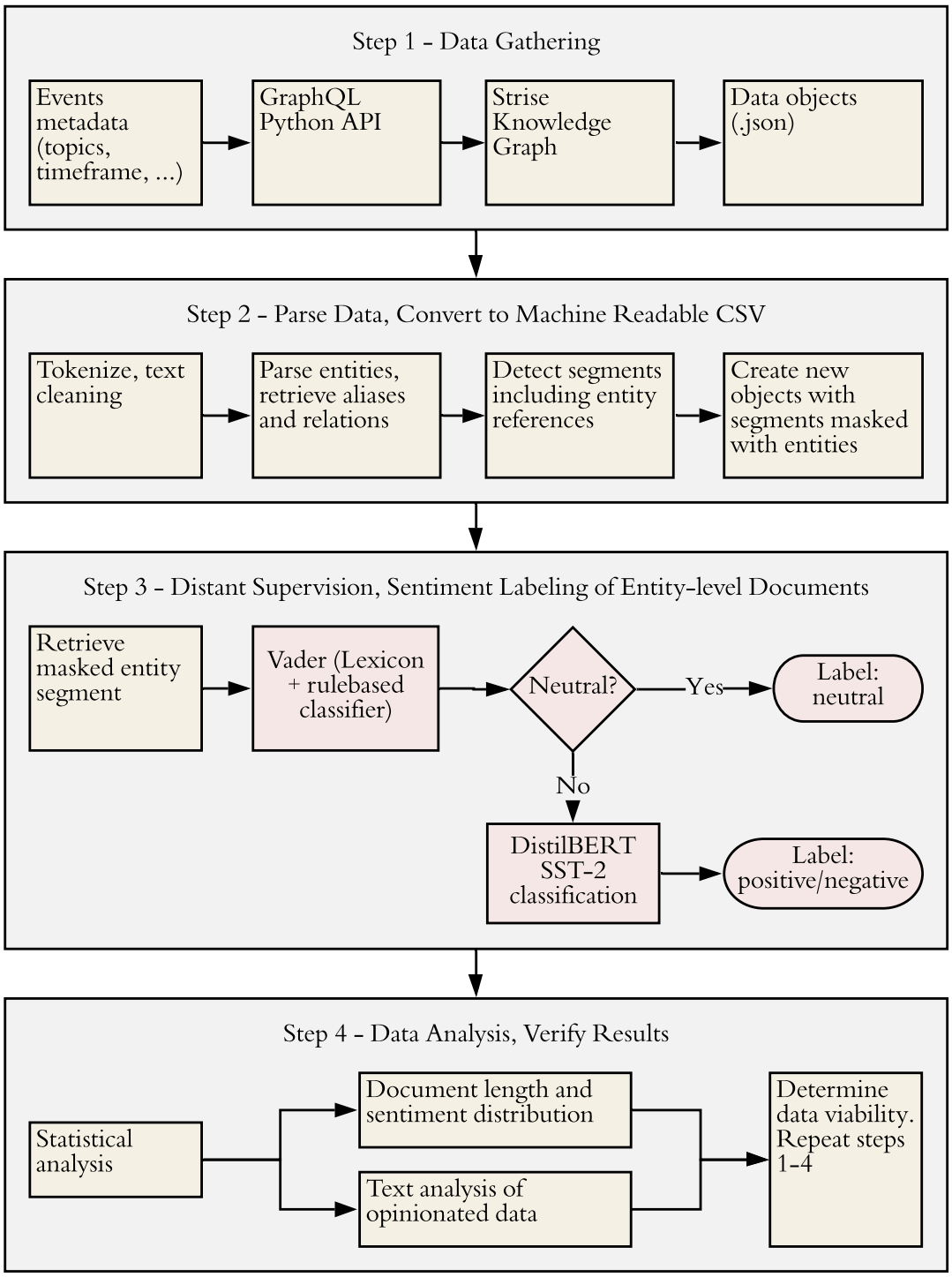


Figure 5.7.: Architecture for dataset creation with data from Strise. Steps 1-4 are repeated with new parameters as needed when the outcome is undesirable.

6. Coreference Validation

Experiments in this thesis, along with their respective results, are divided into two parts: *Coreference Validation* (this chapter) and *Entity-level Sentiment Analysis* (Chapter 7). This chapter lays the necessary foundation for further experiments, involving validation and evaluation of models and modified Coreference Resolution (CR) datasets. First, the setup used for all experiments is presented in Section 6.1, followed by the experimental plan in Section 6.2 which covers the rest of the chapter. Note that both experimental chapters include an experimental plan, set up in such a way that experiments aimed to answer research questions are not mixed between the chapters.

6.1. Experimental Setup

Data used for experiments can be found in the Data chapter, Section 4.4 (page 50). Common for both experimental chapters is the hardware used. Experiments requiring extensive hardware resources were run on the IDUN cluster (Själander et al., 2019) utilizing Dell PowerEdge R630, R730 and R740 servers. Other experiments, typically requiring prototyping, visualization and efficient revision, were run on a personal computer. All hardware can be found in Table 6.1.

Server/Type	**Processor**	**Cores (threads)**	**Memory [GB]**	**GPUs**
Dell PE R630	2x Intel Xeon E5-2630 v2	6 (12)	128	None
Dell PE R730	2x Intel Xeon E5-2560 v4	12 (24)	128	2x Nvidia Tesla P100 16GB VRAM
Dell PE R740	2x Intel Xeon Gold 6132	14 (28)	768	2x Nvidia Tesla V100 16GB VRAM
Personal Computer	1x Intel i7 7700K	4 (8)	16	1x Nvidia GTX 1080 8GB VRAM

Table 6.1.: Hardware used to run experiments

6.2. Experimental Plan

For this chapter, the experimental plan is primarily guided by Research Question 1: "How well do Coreference Resolution models perform when evaluated on out-of-domain data?". The goal to evaluate out-of-domain performance requires setting up models, as well as preparing data, spanning a total of three steps:

1. Reproducibility of Coreference Resolution Models

Set up a selection of CR models and evaluate their reproducibility. This is done to continue on with the rest of the thesis knowing that the models produce expected results. This is presented in Section 6.3.

2. CorefLite Dataset Validation

The resulting CorefLite datasets, covered in Section 4.3.1 (page 47), may be error-prone due to the varying amount of parsing needed in the conversion from the original dataset to the CorefLite format. The converted datasets must be evaluated using the prepared CR models, verifying the integrity of each respective dataset. This process is documented in Section 6.4.

3. Evaluation of Coreference Data

With the models and datasets verified, evaluate each models' performance on the individual datasets, as well as testing the efficiency and other caveats of the models. This must be performed on both out-of-domain and in-domain data, determining model generalizability and news-domain applicability. Results are found in Sections 6.5 and 6.6, which are all original work in this thesis.

6.3. Reproducibility of Coreference Resolution Models

The very first computational experiments conducted on CR included setting up a selection of models discovered in the literature review, to verify their claimed results on the OntoNotes dataset. The heavily distributed e2e-coref model (Lee et al., 2018), SpanBERT model (Joshi et al., 2019a), CorefQA (Wu et al., 2019) and NeuralCoref[1] were prepared. As described in Section 3.2.4, the CorefQA model was missing several required files for setup – as noted by several users on GitHub[2] – and was thus discarded. Moving on with the three other models, some modifications to the core packages of the systems were needed in order to run the models on the IDUN computing cluster – which is strictly a requirement, as all these models require access to excessive computing power. Furthermore, the OntoNotes dataset used for reproducing results is in fact the CorefLite

[1] No paper available, but heavily based around the model by Clark and Manning (2016a)
[2] `https://github.com/ShannonAI/CorefQA/issues/15`

converted OntoNotes. This allows to check for validity for this specific dataset conversion, while simultaneously validating models.

	MUC			B-CUBED			CEAF			CoNNL
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
e2e-coref Reported	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	**73.0**
e2e-coref Reproduced	81.40	79.52	80.45	72.16	69.39	70.75	68.15	67.21	67.68	**72.96**
SpanBERT Reported	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	**79.6**
SpanBERT Reproduced	84.50	82.45	83.46	77.06	75.51	76.27	74.65	72.40	73.51	**77.75**
SpanBERT Pre-trained	81.86	81.55	81.71	73.63	73.64	73.64	71.15	71.13	71.14	**76.12**

Table 6.2.: Reported and reproduced results using the models e2e-coref (Lee et al., 2018) and SpanBERT (Joshi et al., 2019a)

6.3.1. End-to-End Coreference and SpanBERT

The popular *e2e-coref* model by Lee et al. (2018), described in detail in Section 8.2.3 (page 103), has been used as a foundation for several models covered in the literature review. Among these models, the most promising variation is the SpanBERT model (Joshi et al., 2019a). The authors behind e2e-coref were the first to implement a much needed pruning algorithm for the OntoNotes dataset, minimizing unnecessarily large files containing extraneous information for the CR task – which sparked the idea for further minimizing of other datasets for this thesis. Although e2e-coref was considered state-of-the-art at the time of publishing, SpanBERT has improved upon it in every aspect. This has mainly been accomplished by incorporating BERT embeddings (Devlin et al., 2019) modified with span masking (as opposed to single token masking in traditional BERT variations) – outperforming previously used ELMo representations (Peters et al., 2018). The authors of SpanBERT show to a CoNLL F1 score of 79.6, whereas e2e-coref scores 73.0. While the results for SpanBERT could not be entirely reproduced using the provided evaluation scripts, the differences might be due to varying model sizes and possible issues with the reported configurations. The results are, however, quite similar, and indicate that the reported results are legitimate (albeit with other configurations). Regardless, it is of little importance to reach the exact reproduced results, as the performance on non-OntoNotes datasets are far more relevant to answer the research questions. The official results for both e2e-coref and SpanBERT, along with the reproduced evaluations and recalculated

scores from the implementation of a pre-trained SpanBERT model[3] are found in Table 6.2. Due to the vast performance gain from SpanBERT, in addition to it being built directly upon e2e-coref, further evaluations will not be completed using e2e-coref. Furthermore, the team behind AllenNLP (Gardner et al., 2018) developed a Python interface for the pre-trained release of SpanBERT, making it tough to justify developing a custom system to interface with the older e2e-coref model. Model-specific configurations can be found in Appendix E.

	MUC			**B-CUBED**			**CEAF**			**CoNNL**
Deep-Coref/ NeuralCoref	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
Reported Deep-Coref	73.64	65.62	69.40	67.48	56.94	61.76	62.46	58.60	60.47	**63.88**
Computed NeuralCoref	82.08	61.36	70.22	74.13	46.60	57.23	60.11	47.52	53.10	**60.18**
Pre-trained NeuralCoref	71.88	44.86	55.24	64.61	34.21	44.74	55.59	42.56	48.21	**49.40**

Table 6.3.: Reported results by Clark and Manning (2016a), computed results using an implemented and trained NeuralCoref model as well as a pre-trained variation.

6.3.2. NeuralCoref

Contrary to the previous models, no official results are reported on NeuralCoref as of yet, aside from the improvement of ranking loss when compared to the *Deep-Coref* model[4]. In Table 6.3 are the official results on the Deep-Coref model (Clark and Manning, 2016a), alongside the calculated results by training the NeuralCoref model on the OntoNotes dataset, and lastly the results by using a pre-trained release, available through a Python interface[5]. The pre-trained release of the NeuralCoref model seems to be ineffective, especially given that the trained model and pre-trained release should perform similarly – if not identically. To investigate this, hyperparameters were modified using the official API for the pre-trained model. The experiment yielded minuscule improvements, never enabling the model to surpass F1 scores of 0.49 in eighteen passes of the testing dataset (see Appendix D.3). This is unfortunate, as there is currently no way to load a trained model into the NeuralCoref Python interface, without rewriting the entire model to support the CorefLite format. Going by the initial lackluster results, completely rewriting the model was deemed unreasonable for this thesis, and is thus left for future work. The pre-trained model, albeit its apparent weaknesses, provides an extremely fast calculation

[3] https://github.com/facebookresearch/SpanBERT#pre-trained-models
[4] https://github.com/huggingface/neuralcoref/blob/master/neuralcoref/train/training.md#some-details-on-the-training
[5] https://github.com/huggingface/neuralcoref/releases/tag/v4.0.0

of coreference clusters, with the possibility of integrating world knowledge through hyperparameters. For these reasons, the pre-trained model will be included in further experiments.

	MUC			B-CUBED			CEAF			CoNNL
CoreNLP	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
Reported Deterministic	-	-	-	-	-	-	-	-	-	**49.50**
Deterministic	54.86	59.78	57.21	45.98	46.39	46.18	38.26	51.01	43.72	**49.04**
Reported Statistical	-	-	-	-	-	-	-	-	-	**56.20**
Statistical	70.79	63.08	66.71	59.47	47.83	53.02	52.06	46.16	48.93	**56.22**

Table 6.4.: Verifying results using the Deterministic and Statistical models by Lee et al. (2013) and Clark and Manning (2015) on the OntoNotes dataset.

6.3.3. Deterministic and Statistical Models

With the neural results completed, similar evaluations were conducted on the highly regarded deterministic and statistical models by Lee et al. (2013) and Clark and Manning (2015). These are both implemented into the Stanford CoreNLP library (Manning et al., 2014b), and were accessed through Stanford's new Stanza framework (Qi et al., 2020). The importance of evaluating on non-neural models has been established earlier, with Moosavi (2020) stating that larger neural models may perform on-par with rule-based models for out-of-domain evaluation. Moosavi (2020) did not evaluate on more than one smaller dataset (WikiCoref, see Section 4.1.2) – motivating the need to verify with the prepared CorefLite datasets. The CoreNLP Deterministic model (Lee et al., 2013) has been documented in Section 3.2.1, along with a description of its sieve-based architecture (see Table 3.4, page 36). The deterministic model has a reported CoNLL-2012 F1 score of 49.5, and the statistical model 56.2, found on the *Stanford NLP Group* website[6]. Reported and calculated scores on the OntoNotes dataset can be found in Table 6.4, verifying the official results.

6.4. CorefLite Dataset Validation

The datasets OntoNotes, GUM, PreCo and LitBank had to be converted into a unified format, as thoroughly described in Section 4.3.1 (page 47). The conversion proved to be a lengthy process, as there was no other option than to visualize the outcome and inspect individual errors in the resulting clusters. Luckily, whether a document was error-prone or not was easy to determine, as the final evaluation metrics – when using

[6] `https://stanfordnlp.github.io/CoreNLP/coref.html`

Input data	CoreNLP	NeuralCoref	SpanBERT
you	you	you	you
've	've	'	'
done	done	ve	ve
what	what	done	done
you	you	what	what
could	could	you	you
n't	n't	could	could
-	-	n't	n't

Table 6.5.: The tokenized sentence of "You've done what you couldn't" being processed differently across the used models. **CoreNLP** includes both the Deterministic and Statistical model.

wrongly parsed data – would often produce scores nearing nil[7]. Whenever this occurred, the parsing algorithms had to be revisited and revised. The most frequently occurring errors regarded differences in token indices. Observe how the contraction *you've* gets tokenized differently using the evaluated models in Table 6.5. This quickly becomes an issue if a mention cluster includes *you've*, tokenized as [*you*, *'ve*], while the model predicts the span to occur over [*you*, *'*, *ve*]. The offset will modify all forthcoming clusters, and propagate wrong mention indices throughout the document, resulting in highly incorrect mention clusters. The solution was an algorithm executing the following actions:

1. Find indexes where a mismatch between input and output tokens occurs

2. For each mismatched index, if the predicted mention's start/end index are above the given mismatched index, reduce the start/end index by 1

Besides the issues with contractions, the algorithm also handles tokenization issues with hyphens and punctuation – or any diverging tokenization techniques. After thorough testing, the algorithm was added as an action in the prediction pipeline of the CorefLite Evaluation Framework (CL-Eval), and the framework was now ready, allowing for verification of the parsing process. As a last note, some datasets discard singleton mentions – that is mentions not occurring anywhere else, and can thus not be regarded as coreferences. For the CorefLite datasets, no mentions have been filtered (i.e. both singleton and coreferenced mentions are kept), but CL-Eval rather allows for the inclusion of singleton mention as an optional parameter, set by the user. Unless specified, singleton mentions are *not* included in the following results.

[7]The initial CorefLite-version of the PreCo dataset resulted in CoNNL F1 scores between 0.008 and 0.040 using varying models

GUM document candidates	Precision	Recall	F1
Reported (unidentified)	0.6363	0.3835	0.4786
GUM_news_defector	0.6760	0.3857	0.4700
GUM_news_imprisoned	0.6676	0.3693	0.4631

Table 6.6.: Identifying document candidates for verification of the CorefLife-formatted GUM dataset.

	MUC			B-CUBED			CEAF			CoNNL
GUM	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
Reported	57.25	35.22	43.61	50.53	25.64	34.02	39.03	33.18	35.87	**37.83**
Replicated	52.06	35.65	42.32	45.24	23.67	31.08	34.27	31.59	32.87	**35.42**

Table 6.7.: Reported results on the GUM dataset by Zeldes and Zhang (2016), as well as replicated CorefLite results with the CoreNLP Deterministic Model

6.4.1. OntoNotes

As the OntoNotes dataset had already been converted to a similar format through the minimzation process of Lee et al. (2018), only extraneous information was altered. The modified CorefLite OntoNotes was used for all evaluations throughout the initial sections of this chapter, and indicate validity.

6.4.2. GUM

In current literature, only the official paper for the GUM dataset (Zeldes, 2017) and a CR model by the same authors (Zeldes and Zhang, 2016) have published scores on the dataset. The former publication only evaluates one single document (1 out of 120 total), without providing information on evaluation metrics used. Going by an assumption of the undefined metric being the commonly used CoNNL-F1 score, an experiment to identify the unknown evaluated document was conducted. The findings are illustrated in Table 6.6, where two identified candidates show similarities to the published document, and might indicate correlation between the original and parsed dataset. Addressing the latter publication (Zeldes and Zhang, 2016), the GUM dataset was evaluated using the CoreNLP Deterministic model. However, at the time of publishing – June 16, 2016 – the dataset was in an earlier stage. The oldest available data today is dated August 18, 2016[8], and thus, experiments cannot be completely replicated. Nevertheless, results using the August 2016 version (2.1.1) are presented in Table 6.7. Observe that reproduced results are lower than the originally published results, which might indicate dissimilarities between the

[8] https://github.com/amir-zeldes/gum/releases?after=V3.0.0

datasets. However, due to arguable resemblance of the metrics, the conversion process of GUM was deemed valid.

6.4.3. PreCo and LitBank

Neither the PreCo (Chen et al., 2018) nor LitBank (Bamman et al., 2020) datasets have been evaluated by any distributed CR models in discovered literature. As LitBank is rather new, the lack of evaluation is not too surprising. However, the PreCo dataset has been around since 2018, and has still not seen CR evaluations – which is strictly its intended purpose. A possible reason for this may be the unconventional annotation scheme. The CorefLite conversion process, however, was deemed successful upon visual inspection – and thus paves the way for further experimentation alongside the other datasets. Only one available publication shows comparisons of OntoNotes, LitBank and PreCo – from the authors of LitBank themselves. A baseline was set using an undocumented BiLSTM-CRF model (Bamman et al., 2020), originally intended for Named Entity Recognition (NER)[9]. The model was trained on the LitBank dataset, and further evaluated on the test subsets of OntoNotes, PreCo and LitBank, restricting results to the F1 scores. The final results are shown in Table 6.8, and indicate strong performance on the OntoNotes and PreCo datasets if the model is trained on LitBank alone.

LitBank Model Datasets:	**MUC F1**	**B-CUBED F1**	**CEAF F1**	**CoNLL Avg F1**
OntoNotes	57.7	81.2	49.7	62.9
PreCo	63.5	84.2	55.1	67.6
LitBank	62.7	84.3	57.3	68.1

Table 6.8.: Reported results on LitBank and Preco using an undocumented model

A recap The models used for evaluation are CoreNLP deterministic, CoreNLP statistical, NeuralCoref (pre-trained) and SpanBERT (pre-trained). These models are each evaluated on the parsed CorefLite datasets of OntoNotes, GUM, LitBank and PreCo. To further assess domain-specific data, the OntoNotes and GUM datasets, both containing topic identifiers, were modified to split evaluations into the two categories out-of-domain and and in-domain:

- Out-of-domain: OntoNotes (no news), GUM (no news), LitBank, Preco
- In-domain: GUM (news only), OntoNotes (news only)

[9]For more information on these technologies, see pages 18 for Bidirectional Long Short-Term Memory (LSTM), 15 for Conditional Random Field (CRF) and 10 for NER

Note that although the OntoNotes dataset has been stripped of news related data for the out-of-domain category, the NeuralCoref and SpanBERT models are trained on the OntoNotes training set. Some overlap between the train and test set of OntoNotes has been identified (Moosavi, 2020), and thus the metrics on these models are expected to be somewhat biased. Regardless, this data has been included for comprehensive evaluations. Due to the vast number of metrics calculated in the forthcoming experiments, tables from here on are restricted the F1 scores of MUC, B-CUBED and CEAF, including the average of those – the CoNNL F1, and lastly, the LEA metric (Moosavi and Strube, 2016), designed to overcome robustness issues of CR. The LEA metric will be focused for discussions regarding each model and its performance. The complete evaluation data, all metrics included, resides in Appendix G. Highlighted scores indicate the highest value for each metric column. From here on, all results are original work.

6.5. Out-of-Domain Evaluation

Followed are all evaluations in the out-of-domain category. Paying attention to the LEA metric, it seems to penalize low recall – which makes sense when reflecting on the metric's intended purpose: prioritizing the importance of entities and correctness of attributed coreference links (see Background Section 2.4.2, page 27). Observe in the coming tables how the NeuralCoref model consistently discovers more coreference clusters than the CoreNLP Deterministic model, but also fails to associate them with the correct antecedent, resulting in lower recall. The final F1 LEA score reflects the real-world application for this, that being correctly linked coreferences to antecedents, hence it scores the CoreNLP Deterministic model higher than NeuralCoref on all four datasets.

OntoNotes (no news)

In Table 6.9, the results on the modified OntoNotes dataset are shown. As expected, SpanBERT, being trained for this specific dataset, shows the best results. Furthermore, all models perform slightly better on the news-stripped variation, compared to the full size OntoNotes dataset.

GUM (no news)

Moving on to the GUM dataset, things get more interesting in Table 6.10. Although SpanBERT performs better for all metrics, the relative difference between the models is much smaller than for the previous evaluation. Regardless, the SpanBERT model still excels at achieving higher recall scores than its competing models, strengthening its final F1 scores. However, for all models, performance has taken a big hit when compared to the OntoNotes dataset. The GUM dataset consists of web-scraped documents with a diverse vocabulary, and experiments indicate that this diversity may reduce a model's capability for consistently detecting clusters. Furthermore, the effectiveness of neural models (i.e. NeuralCoref and SpanBERT) quickly deteriorate on out-of-domain evaluations, as hypothesized by Moosavi (2020). The SpanBERT model performance dropped 38.8%

Dataset OntoNotes No news	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	59.46	47.37	44.17	50.33	42.54	41.28	41.90
CoreNLP Statistical	68.66	54.10	49.31	57.36	55.49	45.36	49.92
NeuralCoref	56.30	45.55	49.17	50.34	60.81	31.52	41.52
SpanBERT	**83.51**	**74.91**	**73.94**	**77.45**	**72.50**	**72.68**	**72.59**

Table 6.9.: Out-of-domain F1 evaluations + LEA metric on OntoNotes with news documents stripped.

Dataset GUM No news	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	48.00	35.41	35.43	39.61	43.66	22.70	29.87
CoreNLP Statistical	56.29	39.21	32.69	42.73	63.00	24.52	35.30
NeuralCoref	39.52	25.27	26.64	30.48	63.06	13.16	26.64
SpanBERT	**59.00**	**47.66**	**41.73**	**49.46**	**69.79**	**32.59**	**44.44**

Table 6.10.: Out-of-domain F1 evaluations + LEA metric on GUM with news documents stripped.

and the NeuralCoref 35.8% from the OntoNotes evaluation, whereas the deterministic and statistical models dropped 28.7% and 29.3% respectively.

LitBank

The LitBank dataset is vastly different from the other datasets, as documents are much longer – requiring high versatility for a CR model to succeed. As discussed in the Introduction (page 2), long-range dependencies are difficult to resolve. Table 6.11 shows the performance for all models. Observe that all F1 scores are higher for LitBank (with longer documents in formal language) than for the GUM corpus (comprising shorter

Dataset LitBank	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	60.76	37.88	23.61	40.75	37.84	32.42	34.92
CoreNLP Statistical	69.94	41.28	28.66	46.63	49.36	31.87	38.73
NeuralCoref	55.86	31.32	30.24	39.14	53.14	19.55	28.59
SpanBERT	**75.53**	**59.19**	**42.06**	**58.92**	**55.41**	**59.92**	**57.58**

Table 6.11.: Out-of-domain F1 evaluations + LEA metric on the LitBank dataset.

documents with a diverse vocabulary), indicating that all models favor predictable vocabulary over document length. Regardless, SpanBERT produces the best scores for LitBank. The inclusion of BERT embeddings, being pre-trained on a large corpora of text, allows the SpanBERT model to better handle complex semantic structures, giving it a clear advantage over competing non-BERT models. Its CoNNL-F1 score of 58.92 is still lower than what the authors of LitBank reported (Bamman et al., 2020) with a model trained on LitBank and evaluating on its test set – reaching 68.1 (see Table 6.8). However, the model trained on LitBank performs far worse on the OntoNotes dataset – 62.9, where SpanBERT reaches 79.6 in the official publication (Joshi et al., 2019a).

PreCo

PreCo, as described in the Data chapter (p. 45), uses a vocabulary aimed at English speaking preschoolers. This might be the reason for why non-SpanBERT models perform better on PreCo than for LitBank and GUM. Continuing analyzing the LEA F1 metric, observe how the gap between the SpanBERT model and the CoreNLP models is reduced on non-OntoNotes datasets – being completely out-of-domain with respect to the training data for SpanBERT. This is illustrated by the relative performance of models in Table 6.13. The NeuralCoref model performs around 50 to 60 percent of the SpanBERT model, while the CoreNLP statistical model shows up to 88% performance of the SpanBERT model on the PreCo dataset and 79% on the GUM dataset. These results are surprising, as the GUM and PreCo datasets contain vastly different data. Hence, the statistical model proves to be a great alternative for out-of-domain texts – although outperformed in these experiments, leaving SpanBERT the definitive best model for generalized CR. Note that the PreCo *dev* dataset is used, in order to evaluate similarly sized datasets for all presented results.

Dataset PreCo	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	55.22	45.57	44.76	48.52	51.12	33.69	40.61
CoreNLP Statistical	62.61	50.41	46.05	53.02	63.93	36.51	46.48
NeuralCoref	46.53	37.00	41.20	41.57	63.43	22.42	33.13
SpanBERT	**64.09**	**55.71**	**53.82**	**57.87**	**70.05**	**42.48**	**52.89**

Table 6.12.: Out-of-domain F1 evaluations + LEA metric on the PreCo dev dataset.

LEA F1 Relative score	**OntoNotes no news**	**GUM no news**	**LitBank**	**PreCo**
CoreNLP Deterministic	57.7%	67.2%	60.6%	76.8%
CoreNLP Statistical	68.8%	79.4%	67.3%	87.9%
Neuralcoref	57.2%	60.0%	49.7%	62.6%
SpanBert	100%	100%	100%	100%

Table 6.13.: Performance of models on out-of-domain data, relative to SpanBERT on the LEA F1 metric.

6.6. In-domain Evaluation

Extracting news data from the compatible OntoNotes and GUM datasets was done to identify differences in the evaluations when compared to the out-of-domain datasets. The relatively high out-of-domain performance of the CoreNLP Statistical model was previously highlighted. However, for the news domain, this model seems to have reduced performance. Furthermore, the news subsets receive lower scores for nearly all models. Additional evaluations on unmodified OntoNotes and GUM datasets are placed in Appendix G.3.

OntoNotes (news)

The OntoNotes in-domain news subset sees much lower scores than its out-of-domain counterpart, as can be observed in Table 6.14. The performance drop, with respect to the LEA F1 score, is drawn attention to in Table 6.15. The neural models, specifically

Dataset OntoNotes News Subset	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	49.20	42.17	42.45	44.61	34.71	37.84	36.21
CoreNLP Statistical	58.86	48.95	47.80	51.87	54.47	36.92	44.01
NeuralCoref	51.32	41.86	45.31	46.16	56.74	27.34	36.90
SpanBERT	**74.91**	**69.15**	**70.35**	**71.47**	**66.39**	**65.65**	**66.02**

Table 6.14.: In-domain F1 evaluations + LEA metric a news subset of OntoNotes.

LEA F1 score	**OntoNotes out-of-domain**	**OntoNotes in-domain**	**Performance drop (rounded percentage)**
CoreNLP Deterministic	41.90	36.21	14%
CoreNLP Statistical	49.92	44.01	12%
Neuralcoref	41.52	36.90	11%
SpanBert	72.59	66.02	9%

Table 6.15.: Performance drop between in- and out-of-domain variations of OntoNotes.

SpanBERT, show the least negative impact, in addition to outscoring all other models previously documented in the out-of-domain evaluations.

GUM (news)

GUM has shown to be the most difficult dataset for models to perform well on, and this is even more so the case for the news domain, as illustrated in Table 6.16. Note how the deterministic model is the only model performing better on this domain, in fact outperforming the statistical model for both CoNNL-F1 and LEA F1. The increase in performance is denoted in the table as -7%.

NeuralCoref continues to produce poor results. As for the OntoNotes dataset, performance drop is presented in Table 6.17. These numbers differ greatly, and may be a testimony to the varying vocabulary found in the GUM dataset, causing unpredictable results. SpanBERT shows similar results for both variations.

Dataset GUM News Subset	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	47.21	37.56	38.10	40.96	44.02	25.16	32.02
CoreNLP Statistical	47.85	35.21	36.11	39.72	56.88	21.03	30.70
NeuralCoref	37.01	25.41	30.00	30.81	50.64	13.41	21.20
SpanBERT	**57.10**	**47.93**	**45.57**	**50.20**	**60.29**	**34.75**	**44.09**

Table 6.16.: In-domain F1 evaluations + LEA metric a news subset of GUM

LEA F1 score	**GUM out-of-domain**	**GUM in-domain**	**Performance drop (rounded percentage)**
CoreNLP Deterministic	29.87	32.02	-7%
CoreNLP Statistical	35.30	30.70	13%
Neuralcoref	26.64	21.20	20%
SpanBert	44.44	44.09	1%

Table 6.17.: Performance drop between in- and out-of-domain variations of GUM

LEA F1 Score	**OntoNotes In-domain**	**OntoNotes Out-of-domain**	**GUM In-domain**	**GUM Out-of-domain**
CoreNLP Deterministic	36.21	41.90	32.02	29.87
CoreNLP Statistical	44.01	49.92	30.70	35.30
NeuralCoref	36.90	41.52	21.20	26.64
SpanBERT	**66.02**	**72.59**	**44.09**	**44.44**

Table 6.18.: Compared LEA F1 scores on both variations of the OntoNotes and GUM datasets

LEA F1	CoreNLP Deterministic	CoreNLP Statistical	NeuralCoref	SpanBERT
OntoNotes (original)	40.61	48.69	40.50	71.14
GUM	29.87 (73.6%)	35.30 (72.5%)	26.64 (65.8%)	44.44 (62.5%)
LitBank	34.92 (86.0%)	38.73 (79.5%)	28.59 (70.6%)	57.58 (80.9%)
PreCo	40.61 (100%)	46.48 (95.5%)	33.13 (81.8%)	52.89 (74.3%)
Average out-of-domain performance	86.5%	82.5%	72.7%	72.6%

Table 6.19.: Final out-of-domain evaluation table. Percentages indicate the score relative to the OntoNotes (original)

Final Remarks

Reaching the end of domain-specific evaluation, a brief summary is presented in Table 6.18, comparing the two variations of the OntoNotes and GUM datasets. Initial hopes were that the deterministic and statistical model would be able to shine by their rule-based definitions of coreference, enabling higher scores on generally difficult datasets like GUM. This was, on the other hand, not the case, and SpanBERT is left undefeated across all evaluations. Table 6.19 shows the final performance of out-of-domain datasets, relative to the OntoNotes test set. It becomes clear that neural models struggle to a higher degree than the deterministic and statistical models for out-of-domain performance.

7. Entity-level Sentiment Analysis

As the previous chapter concluded, the SpanBERT model was considered the best fit for generalized Coreference Resolution (CR). Experiments in this chapter will utilize SpanBERT to augment datasets for Entity-level Sentiment Analysis (ESA), investigating the impact of CR on this type of data. In a similar fashion to the experiments on Coreference Validation (Chapter 6), another experimental plan is presented below – directed by the so far unattended research questions. Hardware used to conduct the coming experiments is the very same as for the previous experiments (Table 6.1 (p. 73).

7.1. Experimental Plan

The individual steps contained within this plan take part in resolving the two remaining research questions:

Research question 2 *Can current datasets for Entity-level Sentiment Analysis be used as out-of-domain evaluation baselines?*

Research question 3 *Can augmentation of datasets result in improvements using Entity-level Sentiment Analysis models?*

1. Baselines and Coreference Augmentation

Establish a baseline on existing datasets for ESA in order to have legitimate results for future comparisons of models. Moreover, augment these datasets with CR and study the results. Documented in Section 7.2.

2. Evaluate Generated Data

With the generated dataset from the Distant Supervision (DS) process, employ CR to augment the data and perform evaluations using models for ESA. Furthermore, compare the results to the previously established baselines and utilize existing datasets in experiments. These evaluations are found in Section 7.3, and aim to resolve research questions 2 and partly 3.

3. Manual Annotation

To achieve realistic results – and to further clarify research question 3 – manually annotate a selection of articles extracted from Strise evaluation data. By training a model on available data, evaluate using the manually annotated data. This may help for robust testing of possible use-cases for CR. Findings on manual annotation reside in Section 7.4.

7.2. Baselines and Initial Coreference Augmentation

Before establishing baselines, the Entity-level Sentiment Analysis Framework (Elsa-Val) had to be set up for evaluating several models and datasets. The datasets used are SemEval 2014 – Task 4 (Pontiki et al., 2014) and ACL-14 (Dong et al., 2014), as chosen in the Data chapter, p. 50. The SemEval 2014 task includes two datasets, containing laptop and restaurant reviews. The ACL-14 dataset contains Twitter data. Therefore, the names *Laptop*, *Restaurant* and *Twitter* will be used when referring to the separate datasets. Continuing, the selected models (described in detail in the Architecture of Elsa-Val, from p. 68) are an LSTM, Target-Dependent LSTM (TD-LSTM), Content Attention (CABASC) and Local Context Focus with BERT (LCF-BERT).

Selecting Hyperparameters

Hyperparameters drastically affect the results of a machine learning model, and are often tuned by extensive testing – commonly referred to as a *hyperparameter optimization*. This process was not deemed necessary for this thesis, however, as the main goal is to study the impact of augmenting input data with Coreference Resolution (CR), rather than finding the best possible scores for Entity-level Sentiment Analysis (ESA) in general. Therefore, the selection of hyperparameters was guided by recommended values found in the models' publications. A primary divergence from default parameters was on the LCF BERT model, initially using 16 as its batch size, but this was reduced to 12 in order to complete experiments on all hardware available. See more on batch sizes and other hyperparameters in Background, p. 22. Furthermore, an experiment was conducted to tune down the necessary number of epochs required, to find a balance between satisfactory results and time efficiency. These first tests allowed 20 epochs for the LSTM, TD-LSTM and CABASC models, and – as the authors claim that the model generally converges within three epochs (Zeng et al., 2019) – 5 for LCF BERT. Table 7.1 shows the computed epochs for reaching convergence with respect to the F1 metric. Based on these results, as well as adding model-dependent computational time as a factor (LSTM being the fastest, LCF BERT the slowest), the final hyperparameters were set, defined in Table 7.2.

A Note on Epochs

If a model converges on epoch 3 out of 20, this may indicate **1)** that the model overfits to the training data from this point, or **2)** that the model simply cannot find patterns in the data, resulting in the best model occurring at an arbitrary epoch. Ideally, the

Model	**LSTM**	**TD-LSTM**	**CABASC**	**LCF BERT**
Max epochs	20	20	20	5
Laptop	14	13	4	3
Restaurant	11	8	7	3
Twitter	7	6	17	2

Table 7.1.: Number of epochs required to reach the best F1 scores on the test set for each model on existing datasets.

Model	**Dropout**	**L2-regularization**	**Batch size**	**Learning rate**	**Epochs**
LSTM	0.2	0.01	3	0.001	20
TD LSTM	0.2	0.01	3	0.001	15
CABASC	0.1	0.01	5	0.001	10
LCF BERT	0	0.0001	12	0.00002	5

Table 7.2.: Entity-level Sentiment Analysis Hyperparameters

	LSTM		TD-LSTM		CABASC		LCF BERT		Avg F1 Diff (%)
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	
Laptop	67.87	63.22	67.08	61.22	67.87	61.89	**76.96**	**72.91**	
+ CR	**70.69**	**66.06**	**67.87**	**61.91**	**69.75**	**64.12**	76.49	70.68	+ 3%
Restaurant	**77.05**	**65.05**	**75.71**	**66.55**	75.80	**64.82**	83.30	74.63	
+ CR	74.82	63.43	74.91	62.08	**76.07**	63.93	**84.02**	**76.20**	- 2%
Twitter	67.20	65.33	**69.51**	65.91	64.02	61.68	**73.55**	**72.15**	
+ CR	**67.77**	**65.72**	67.20	**66.03**	**65.17**	**63.26**	70.95	69.39	- 0.25 %

Table 7.3.: Baselines for select models with and without CR augmentation. For each dataset, boldface items indicate the highest score for each metric.

best model should occur towards the end of the number of epochs – meaning that longer training produces better results, indicating that certain semantic and syntactic patterns have been detected, corresponding to the labeled sentiment polarity. This is far from the case. Aside from a few instances, no real correlation between number of epochs and the best performing models was found. This will be further tested in the coming sections.

Preliminary Results

In Table 7.3 are the results from evaluating the aforementioned datasets and models. Boldfaced numbers indicate whether the augmented data (denoted as +CR) or the

original data scored the highest for each metric. These first results are reminiscent of the evaluation of CR models in the previous chapter – that being the BERT model excelling, producing higher scores than its competitors. Somewhat surprising are the high values for the LSTM implementation. Basic LSTM models are often left out for evaluations in newer publications (Liu et al., 2018; Rietzler et al., 2020; Zeng et al., 2019), although the LSTM model outscores both the TD-LSTM and CABASC models in several cases. The TD-LSTM model was only officially evaluated on the Twitter dataset (Tang et al., 2016), which is the only dataset it performs better on than the model it was developed to improve, namely the LSTM. What is less surprising, however, is the lack of impact from CR. As previously discussed, these datasets contain short documents, making it less likely to discover coreference links within them, and thus the theoretical maximum gain from a CR model is low to begin with[1]. Upon closer inspection, the CR augmentation added an additional 7.5% of documents for the Laptop dataset and 5.5% for the Restaurant dataset. These two percent points may be the factor causing the average F1 score difference of 5% when applying CR. With these baselines on official datasets established, the same procedure may begin for the generated dataset.

7.3. Evaluation of Generated Data

The generated dataset annotated by Distant Supervision (DS), thoroughly documented in the Data chapter (pp. 52-56), contains more than 47,000 documents (after segmenting full-text events) evenly distributed over the topics *Technology*, *Sports*, *Politics* and *Business*. This dataset, following the same process as above, was evaluated using the models within Elsa-Val. Two separate *splits* were set up, allowing more detailed evaluations. More information on the splits can be found in Table 7.4.

Train/test split The data was split in an approximate 9:1 ratio. This was accomplished by first batching the segments on the ID corresponding to the source text, then assigning a random selection of 10% of the unique IDs to the test set. This approach ensures that no overlapping data occurs between the two sets. Stratifying on the sentiment scores with this approach is troublesome, as one ID has a varying number of segments within it. The random selection, however, consistently produced the desired distribution of sentiment scores for the segments.

Time split As emphasized by Liu (2017), the time aspect of sentiment analysis is important, as language evolves over time (e.g. by *lexicalization*). This is especially reflected in news-like data. Unique for 2020, with the Corona pandemic affecting all aspects of online media – no matter the topic – results are expected to be lower than for the initial train/test split. To examine this time-specific phenomenon, the training set included data from 2018–2019, and all data from 2020 was included in a separate test set.

[1] More information on the data and analyses can be found in the Data chapter, p. 50

	Training data #segments	Test data #segments	Training data file size [MB]	Test data file size [MB]
All data	43116	4151	19	1.9
2018-2019	40734	6584	18	3

Table 7.4.: The two dataset splits used for experimentation with Distant Supervision.

	LSTM		TD-LSTM		CABASC		LCF BERT		Avg. F1
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Diff (%)
DS all data	**67.67**	**62.80**	62.64	**59.72**	65.31	55.83	**76.44**	**74.58**	
+ CR	67.14	61.86	**65.24**	57.52	**66.01**	**56.86**	73.86	70.45	- 2.5%
DS 2018-2019	**64.09**	**59.97**	**62.65**	**55.88**	**62.32**	**52.38**	**74.23**	**71.73**	
+ CR	63.65	59.82	59.34	55.34	61.88	51.48	71.80	69.33	- 1.7%

Table 7.5.: Model performance on the generated dataset, with and without CR augmentation.

In Table 7.5 are the experiments on the generated dataset. There is an evident drop in performance when evaluating older data (2018–2019) on newer data (2020), confirming the need for time-relevant data for Sentiment Analysis (SA) and other Natural Language Processing (NLP) applications.

7.3.1. Revisiting Hyperparameters

Before continuing evaluating additional data, the hyperparameters set earlier were revisited. Focusing on the number of epochs, they were once again evaluated in an experiment to check for early convergence, with the goal to reduce computational time. Observe in Table 7.6 how the number of epochs vary greatly between the datasets – although the previously set maximum number of epochs seems to fit the results nicely. The number of epochs for the LSTM model could arguably be lowered, but due to its short computational run time, it was left as-is. Unfortunately, with the current setup for CR, few positive results are observed. The earlier hypothesis on the augmentation causing increased entropy within the data, fracturing the original documents, might have proven to be correct. Disregarding the lack of positive impact from CR, however, model performance is rather high. The allegedly good results may not be too surprising, though, given that another sentiment analysis model has in fact labeled all this data, and the respective models used in Elsa-Val may have managed to spot patterns in the labeling process itself. These results are not complete, as they do not represent data rooted in any established truth. To accomplish a proper evaluation scheme, existing datasets will be used.

#Epochs Best F1 score	**LSTM**	**TD-LSTM**	**CABASC**	**LCF BERT**
Max epochs	**20**	**15**	**10**	**5**
Laptop	14	13	4	2
+ CR	11	13	3	1
Restaurant	11	8	7	5
+ CR	12	13	6	5
Twitter	7	6	5	2
+ CR	15	11	7	5
DS all data	12	8	7	1
+ CR	11	14	7	1
DS 2018-2019	6	12	4	2
+ CR	12	2	10	2

Table 7.6.: Number of epochs required to reach the best F1 scores on the test set for each model on original and augmented datasets.

7.3.2. Existing Data as Evaluation Baselines

As previous results merely indicate that augmentation with CR has negative results on the generated dataset, it is necessary to evaluate on other, real world data. For this, existing data previously used (Laptop, Restaurant and Twitter) will be set as baselines, before evaluating on the respective training and test sets (i.e. all available data). This experiment will not only allow for an applicability test of the generated dataset, but will also reveal whether existing data as test sets, although completely out-of-domain, can give any meaningful results using the involved ESA models. Presented in Table 7.7 are the final results, indicating that there is in fact some use for performing these out-of-domain evaluations. The Twitter dataset show to be largely incompatible with the labeled data. However, pay attention to the rows where DS is evaluated on the Laptop and Restaurant testing datasets, diverging only 15 and 17 percent from its original score. The F1 scores of 61.41 for the Laptop dataset and 63.15 for Restaurant, are both similar to what the LSTM, TD-LSTM and Cabasc models reported on their associated training datasets (i.e. all humanly labeled data), as previously seen in Table 7.3. This dictates that the DS approach has somewhat managed to mimic the behavior of properly annotated datasets – which will be tested further in the next section.

	LCF BERT		**F1 difference, original data**
	Acc	F1	
Laptop train Laptop test	76.96	72.91	
DS Laptop test	64.58	61.41	-15.77%
DS Laptop train	65.08	59.15	-18.87%
Restaurant train Restaurant test	84.02	76.20	
DS Restaurant test	73.12	63.15	-17.13%
DS Restaurant train	66.49	58.92	-22.67%
Twitter train Twitter test	73.55	72.15	
DS Twitter test	43.35	43.32	-39.96%
DS Twitter train	42.65	42.66	-40.87%

Table 7.7.: Accuracy and F1 scores for the LCF BERT model when using existing datasets for entity-level sentiment analysis as evaluation baselines for out-of-domain data. The notation in the rows for the leftmost column shows two stacked datasets, with the training set on top, and testing set below.

7.4. Manually Labeled Data

From the selection of evaluation data (see Section 4.5.1, p. 52), articles were extracted, then manually annotated with entities and their respective sentiment. This was done in collaboration with an employee at Strise – Alf Jonassen, to verify the labels by inter-annotator agreement. The process required meticulous work, as isolating meaning towards different targets within a text proved to be far more difficult than first imagined. It is no surprise that the machine learning models struggle with the very same distinction on longer documents. The labeling process was approached as follows:

1. Extract articles
2. Select only articles with text lengths between 800 to 2000 characters to uphold some consistency in the data
3. Sample 25 articles from each topic (100 total)
4. Evaluate each article separately, selecting (if possible) two unique entities within the same text, preferring them to be of differing sentiment polarity
5. Agree on the expressed sentiment polarity towards each entity
6. Add one copy of the text for each entity, masking it as T (target) and append its polarity.

For continued results, only the LCF BERT model is used. Although it is the larger model, it generally produces better results in less time than any of the other models, as quickly discovered throughout extensive experiments. Moreover, it outperforms the other models by a large margin, making it the obvious choice for narrowing down the model selection scope.

	LCF BERT		CR F1 Diff (%)
	Acc	F1	
Distant Supervision (all data)	**47.46**	43.43	
+ CR	**47.46**	**43.81**	+ 0.0087%

Table 7.8.: Initial results by training on the generated dataset, testing on the Gold dataset

7.4.1. Initial Results

After processing the first 30 articles, resulting in 59 labeled documents with differing entities, the first experiments were conducted. In Table 7.8 are the inital results when evaluating the generated dataset on the manually annotated dataset (hereafter referred to as *Gold*), with and without CR augmentation. The results were disappointing, both in terms of evaluation performance and with regards to the impact of CR. The poor results, however, sparked an idea of revising parts of the augmentation hypothesis.

Best model 5 epochs	Laptop		Restaurant		Twitter		Distant Supervision (train/test split)	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Original Train/Test	76.96	72.91	83.30	74.63	**73.55**	72.15	**76.44**	**74.58**
CR Train Original Test	76.49	70.68	**84.02**	**76.20**	70.95	69.39	73.86	70.45
Original Train CR Test	**77.26**	**73.04**	83.46	76.04	73.45	**72.22**	71.12	67.87

Table 7.9.: Different combinations of augmenting train and test data for the existing datasets, as well as the distant supervision dataset.

Best model 5 epochs	Laptop			Restaurant			Twitter		
	Acc	F1	STD	Acc	F1	STD	Acc	F1	STD
Gold	40.68	32.78	3.75	49.15	44.06	5.08	**38.98**	25.78	**4.13**
+ CR	**44.23**	**43.11**	**2.20**	**53.85**	**51.36**	**4.22**	36.54	**31.41**	6.06
CR Impact (rounded %)	+ 8%	+ 24%	- 71%	+ 9%	+ 14%	- 20%	- 7%	+ 18%	+ 33 %

Table 7.10.: Evaluations on Gold data using existing Entity-level Sentiment Analysis datasets as training data. STD is the standard deviation of the computed F1 scores over the five epochs. A low number is desired, as it implies lower variance in the predictions.

7.4.2. Revising the Augmentation Approach

From the initial results, the generated dataset has proved to be of little use for additional evaluations on the Gold dataset. Therefore, instead of augmenting the training data – causing apparent inconsistencies – the *test data* was instead considered for augmentation. The first experiments using augmented test sets (Table 7.9) were initially discarded, as there was seemingly nothing to extract from the results. However, this occurred before considering reusing the existing datasets (Laptop, Restaurant and Twitter) as *training data* for other experiments, where the manually annotated dataset was augmented and used as a test set. With this approach, all data would be labeled by humans, hence the results would become more legitimate. The results in Table 7.10 show that the augmented test set receives far higher scores than in its original non-augmented state, with much less variance in the data (as represented by the STD column). The improvements using the Laptop and Restaurant datasets are rather high, while the Twitter dataset is evidently

Best model 5 epochs	Laptop ∪ Restaurant		
	Acc	F1	STD
Gold	44.07	35.33	4.55
+ CR	**51.44**	**50.23**	**3.46**
CR Impact (rounded %)	+ 14%	+ 30%	- 32%

Table 7.11.: Evaluations on the Gold dataset using the combined data from SemEval 2014 – Task 4 (Pontiki et al., 2014)

less useful for this task (further confirming earlier discussions on the text length being vital). The Laptop and Restaurant datasets show the best scores. For a final experiment, these were joined, combining both their individual training and testing datasets to study the possible outcome on the Gold dataset. This is shown in Table 7.11. The results are promising, and also indicate that the Restaurants dataset was the most appropriate data source for the current data residing in the Gold dataset – when comparing the previous results (Table 7.10) to the combined dataset (Table 7.11). Nevertheless, both the Restaurant and the combined dataset outperform the generated dataset substantially, which was quite surprising – indicating that carefully labeled data (albeit out-of-domain) far outperforms massive amounts of automatically labeled data for these specific tests.

8. Evaluation and Discussion

Having presented experiments and results, there are several standing questions to be settled. Starting with the research questions, each will be answered in turn, before a final evaluation of the main goal for the thesis. Followed are some thoughts and discussion on the fields of Entity-level Sentiment Analysis (ESA) and Coreference Resolution (CR), before a final discussion on the generated dataset.

8.1. Evaluating Research Questions and the Main Goal

Below are the three defined research questions, followed by the overall goal for the thesis.

Research question 1 *How well do Coreference Resolution models perform when evaluated on out-of-domain data?*

This question eventually became the very basis for the thesis, requiring much more attention than initially presumed. Nevertheless, it turned out to be the most rewarding one to answer at that, laying foundations for future research in the field of CR through the work published on out-of-domain evaluations and conversion of datasets.

Results show that CR models take an excessive hit when faced with out-of-domain data. The SpanBERT model (Joshi et al., 2019a) proved to be the most generalizable CR model to out-of-domain datasets, but still has a long way to go. For instance, performance dropped as much as 37% from the reported scores on the OntoNotes dataset when evaluated on the GUM dataset. The undefeated performance of SpanBERT, however, is of little surprise, as variations of BERT models have excelled in a vast majority of Natural Language Processing (NLP) tasks in the past year (Radford et al., 2019), and CR seems to be no exception. With the random masking of spans, SpanBERT has proved to be effective in identifying the inherent semantic and syntactical properties contained within these spans of text – and is clearly a successful modeling approach.

But what is implied by the *performance* of a model? This is somewhat hard to define. Performance of a CR model depends on so much more than just numbers, and is rather based on the specific desired outcome for a system – heavily dependent on empirical evaluation. Specific tasks can involve only resolving coreferences for companies, certain topics and predetermined persons. Modeling this in advance is frankly impossible, and thus the presented evaluations are only a guideline for how the selected four models perform on the tested data. If predictability is desired across different data sources, the deterministic model by (Lee et al., 2013) is perhaps preferable, achieving the lowest

variance in the predictions on out-of-domain dataset evaluations (p. 87). As concluded, however, for general-purpose data, SpanBERT is the best choice.

Research question 2 *Can current datasets for Entity-level Sentiment Analysis be used as out-of-domain evaluation baselines?*

As briefly discussed in Section 7.3.2 (p. 94), results on out-of-domain baselines show that the existing Laptop and Restaurant datasets could be used for evaluations on the dataset annotated by Distant Supervision (DS). This is an important finding, as it enables automated approaches to use existing annotated data to verify the performance generated datasets, thus the development of datasets can be altered according to the evaluated results. Alteration must be approached with care, however, as tuning the DS techniques to give optimal results on out-of-domain data, may ultimately result in abysmal performance on data for the desired application. As originally presented in Table 7.7 (p. 95), results on the training sets are considerably lower than for the tests. This indicates that larger amounts of out-of-domain data (i.e. training sets) will eventually reduce the final scores. An idea to handle this is presented in Future Work, Section 9.3.2, with an example of a metric to include this difference between training/test sets as a heuristic for evaluation purposes.

Research question 3 *Can augmentation of datasets result in improvements using Entity-level Sentiment Analysis models?*

Final results do not give a clear understanding of the benefits of dataset augmentation. The initial approach of augmenting training data to give the models more input to learn from proved to be inefficient, causing increased entropy in the data in most cases, negatively impacting the involvement of CR. Although the Laptop dataset saw an average increase of 3% by augmenting its training data, the LCF BERT model on the same dataset gave undesired outcomes. This indicates that the process was not robust, nor reliable.

To further study the task, it was attacked from another angle by augmenting test data – hypothesizing that models would be able to better predict segments of data, rather than the full texts. This approach, using the Laptop and Restaurant datasets as training data, yielded significant improvement of accuracy and F1 scores when testing on the gold dataset – also reducing the standard deviation of the produced results. Four confusion matrices are presented, including the recently discussed data – trained on LCF BERT. Figure 8.1 represents the last results found in the experiments chapter (p. 98), along with the baselines for the original Laptop and Restaurant datasets, merely as a visual reference guide for desired results in a 3×3 confusion matrix. The labels positive, neutral and negative are represented by ☺, 😐 and ☹, respectively. A perfect classification would show red values along the diagonal, leaving everything else white. Orange vales indicate a majority of correct predictions, moving down through yellow, and white finally indicates zero correct predictions. The CR augmentation shows improvement, enabling

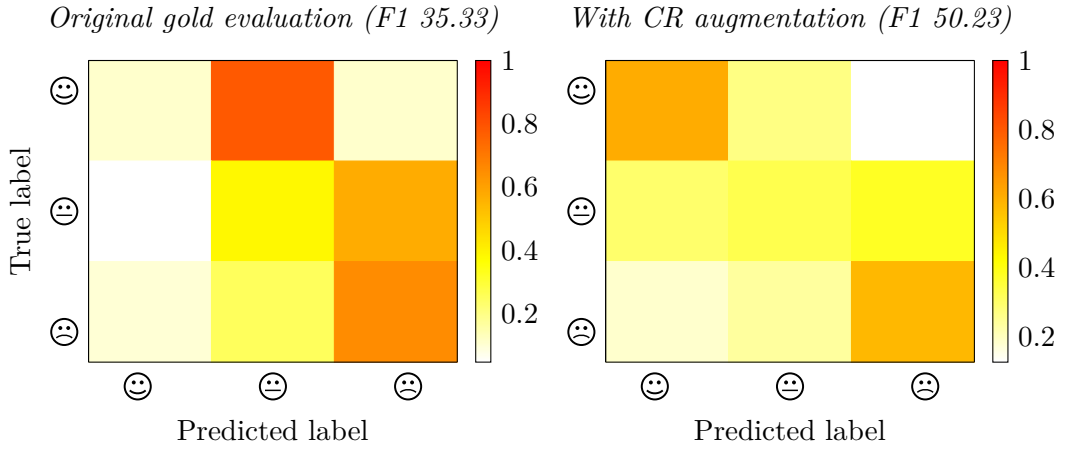


Figure 8.1.: Confusion matrix for Gold evaluation dataset and coreference augmentation.

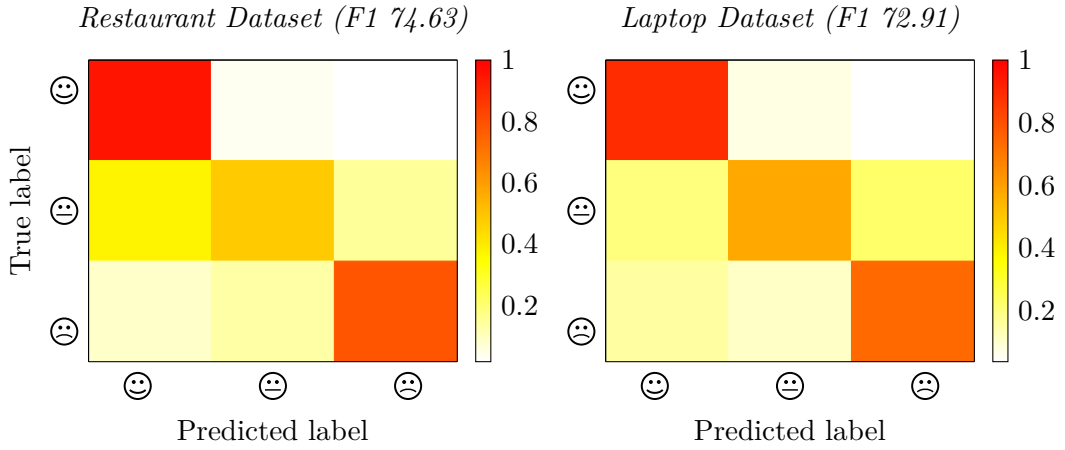


Figure 8.2.: Confusion matrix for Restaurant and Laptop baselines

the ESA model to predict true positive values at a much higher rate (increased by 60 percent points). However, values for true negatives and true neutral are marginally worse, losing 7 and 5 percent points. Irrespective, the final F1 score is much greater for CR – with higher overall precision and recall of correctly classified documents – and thus the entity-centric segmentation algorithm turned out to be valuable when applied to test data. Finally, addressing the research goal:

Goal *Establish a well-generalized Coreference Resolution model to augment the task of Entity-level Sentiment Analysis*

A well-generalized CR model has been established, evaluated on a great variety of data. As discussed earlier, SpanBERT is one of many models based on the work by Lee et al. (2018). It could be argued that other models with a similar foundation would produce

similar results to the SpanBERT evaluations – which cannot be verified without additional testing on the specific models. The reasoning behind selecting SpanBERT, however, is due to its performance on the OntoNotes dataset being substantially greater than for its preceding models (see Table 3.7, p. 41). From the studies conducted, not to mention the indisputable performance gains from BERT models in recent times, it can be stated with confidence that SpanBERT is the model of choice for domain-independent, generalized CR.

Assessing the latter part of the goal, the augmentation of ESA has turned out to be somewhat vague, as discussed above for Research Question 3. The end result, after revising the hypothesis, was discovered quite late – leaving few resources left for augmentation of additional test data. Furthermore, note that augmenting the test data may not be a worthy approach for some applications, such as those regarding shared tasks and other competitions, as discovered through the manipulation of this data in previous experiments (Table 7.9, p. 97). The interest for this thesis lies in researching real world applications, especially involving longer articles. For instance, if a new document arrives into a system, it should be able to manipulate it as much as needed in order to produce the best results for applying in-depth text analysis. Results show that augmentation of previously unseen data has been effective, motivating further research in the fields of ESA and CR as a joint system.

8.2. Discussion

The two first sections concern the task of Entity-level Sentiment Analysis (ESA), followed by a discussion on the constraints of Coreference Resolution (CR) and use cases for generating datasets.

8.2.1. Spotting Patterns in Overlapping Data

From the presented results on ESA through various experiments, it can be tough to discern between good and bad values, as plenty of them are found in similar ranges (often occurring between 60-70). Regardless, LCF BERT consistently performed better than the other models, but why? One underlying reason is the BERT implementation and the transformer architecture.

Initially caused by a mishap, data was generated in such a way that an overlap existed between the segments found in training and testing data. This resulted in the testing data containing similar language to that found in training data, and in some cases the very same text, only with different entities masked as the unknown entity to be classified (T). Consequently, some of the overlapping texts would be written by the very same author. When evaluated using the previously set hyperparameters (p. 93), results showed that the LCF BERT model was reaching exceptionally high accuracy and F1 scores. To further investigate how much of the overlapped data the models were able to learn from, the number of epochs was set to 20 for LSTM, TD-LSTM and

CABASC, and 15 for LCF BERT. This was a quite extensive experiment (in terms of computational time), but returned interesting results – presented in Table 8.1. Note how LCF BERT did not reach convergence before it was stopped at 15 epochs for both datasets. The model has clearly been able to detect a pattern between the noise of masked entities, whereas the other models still struggle with this task, producing similar (and in some cases worse) results compared to the non-overlapping data (Table 7.5, p. 93). Concluding, the results do not represent any real world translatable applications, but are an example of the importance of non-overlapping data, as well as illustrating the remarkable effectiveness of pre-trained language models.

	LSTM			**TD-LSTM**			**CABASC**			**LCF BERT**		
	Acc	F1	Epoch	Acc	F1	Epoch	Acc	F1	Epoch	Acc	F1	Epoch
Overlap Train/Test	67.92	62.82	4/20	64.39	60.57	7/20	62.27	52.96	20/20	**98.82**	**98.87**	**15/15**
+ CR	65.97	55.64	16/20	64.20	54.72	11/20	62.74	52.19	19/20	**98.74**	**98.64**	**15/15**

Table 8.1.: Overlapping data and scores on different models, alongside the number of epochs needed to produce the best model. The boldface is added to emphasize the extremely high values for the LCF BERT model.

8.2.2. Issues with Current Annotation and Modeling Schemes

Current models are set up to handle the entities or aspects within a text individually. This is the biggest flaw uncovered, regarding efficiency of both the data annotation structure and modeling approach. Models have to process the same input N times for N entities occurring in the text in current systems – a number which can quickly increase along with the text length. Caching could resolve some problems, but is hardly an elegant solution, as the texts contained in the datasets are still multiplied for each entity. The transformer architecture, however, as described in Background (p. 21), could be utilized to handle multiple entities as input. Some thoughts on proper usage of transformers are presented in Future Work, Section 9.3.1, diverging from that found in existing BERT solutions (Rietzler et al., 2020; Zeng et al., 2019).

8.2.3. Re-implementation and Code Butchering

The model by Lee et al. (2018) has clearly seen great traction in the development of recent CR models. It has served as the backbone of the models proposed by Subramanian and Roth (2019), Fei et al. (2019), Joshi et al. (2019b), Joshi et al. (2019a), Wu et al. (2019) and Zhang et al. (2019) and will likely see heavy usage in coming research. A quite severe problem with the indisputable re-implementation of other systems has resulted in irreproducible and inconsistent code for the aforementioned models – i.e. *butchering* of the source code. In the process of determining the quality and legibility of published code, several similar files were found, with tiny changes between them. At first glance, this may not seem like a problem, but as systems become open source, *some* of the code-bases

may see updates over time, as valiant contributors handle bugs, tune parameters and make other improvements to the systems. In an example where six other models (such as those above) rely on code published by a seventh, e.g. the model by Lee et al. (2018), and any of the seven systems change its code, there are six outdated code-bases – some of which may be the new source for forthcoming research. For this reason, it is with great importance for future researchers to improve their coding guidelines, updating licenses and documentation, as well to utilize symbolic links to reused code. In such a way, coming systems may be reused with a clear understanding of where any given file was first conceived, avoiding further mishaps.

8.2.4. Unleashing Coreference Resolution

The end result of this thesis, being the augmentation of documents, has primarily regarded the three ESA datasets, comprising online reviews and Twitter posts, a generated dataset on news items as well as a manually labeled dataset on the same type of data. Throughout the Data chapter, the correlation between text length corresponding to the number of CR clusters was discussed, concluding with CR having the most prominent effect on longer documents. It can, therefore, be argued that CR is well suited for the news domain. However, there is a constraint on how a CR model can prove its worth when aimed at augmenting predetermined entities.

Constraints of Predetermined Classification After the experiments on CR were conducted, the observed value of CR was clearly held back by the constraints of the data for ESA. CR models can detect additional details of the input, not labeled within any dataset for ESA, which leaves the model constrained to look for coreferences to the target entity already defined (in order to improve the chances of correct predictions) – which is the case for this thesis. This is impossible to circumvent when aiming to produce numerical results, but the possible use case for CR expands far outside the limits of predetermined annotations and datasets.

A Real World Application Intuitive and observed value of a system is hard to present scientifically, as there is no way to report the findings outside visualizations. By looking up an arbitrary article (original source from TechCrunch[1]), the difference between constrained and non-constrained CR is tremendous – illustrated in Figures 8.3 and 8.4. The text may be illegible, but the important part is that each unique entity is represented by the same color. By utilizing unconstrained CR, the real world application for augmenting ESA becomes much more attainable. Such a wide range of augmentation requires extensive rewriting of current systems, and is left for Future Work in Section 9.3.4.

[1] `https://techcrunch.com/2020/05/11/elon-musk-restarts-tesla-factory-in-defiance-of-county-orders/`

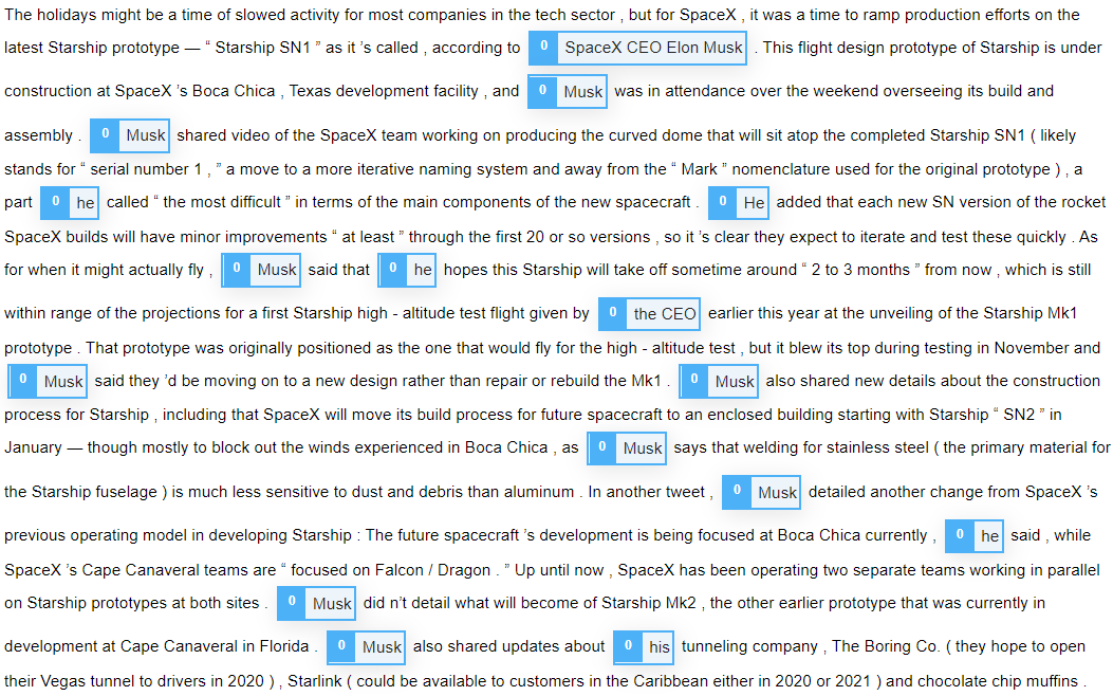


Figure 8.3.: An example of when a Coreference Resolution model is constrained to an entity.

Figure 8.4.: An example of when a Coreference Resolution model is unconstrained.

8.2.5. The Generated Dataset

Early in the research phase, the value of a generated dataset was perhaps overestimated. The leading hypothesis at the time was that a massive dataset would, in good faith, return significant results for both CR and ESA. Looking back, this train of thought was rather naïve. Although the dataset did produce somewhat passable results for the ESA task (p. 95), the end result left much to be desired. The novelty of the procedure must be emphasized a little, though. By including aliases and relations from the knowledge graph, a large set of additional information about each entity was easily available. A key factor was the selection of which relations to include. This was a heavily iterative, error-prone and time-consuming process, and an optimal solution can never truly be found (without a preset specification of the desired results). The final dataset had to be selected purely on visual inspection from a random selection of the data, as manually inspecting over 47,000 texts for every change in the system is effectively impossible. Thus, the definition-of-done was simply set as *good enough*. Having said that, this approach has turned out to include far too many uncertainties to soundly base future development on. Its use case for text mining tasks may very well be valid, especially the incorporation of world knowledge – as suggested by Ferreira Cruz et al. (2020), but not so much for automatic annotation.

9. Conclusion and Future Work

Wrapping up the thesis, a conclusion is presented, followed by contributions and suggestions for future work. This has been an exciting journey, leaving behind a high interest to keep studying the fields in the coming years.

9.1. Conclusion

This thesis has presented two larger experimental chapters, with the goals of 1) defining a generalized Coreference Resolution (CR) model, and 2) evaluating Entity-level Sentiment Analysis (ESA) models with and without augmentation by CR. Extensive evaluation has proven to be essential in order to determine robust and generalized models for CR, as results on one domain do not necessarily translate well to another. To rectify the lack of thorough evaluations in current literature, three other datasets have been converted to a unified format, enabling evaluations on a much broader level than previously possible. Experiments show that a model based on pre-training and Neural Networks (NNs) produces the best results on the tested datasets – defining the most desirable model for augmentation tasks. However, the performance of NN-based models quickly diminishes when evaluated on previously unseen data, with an average reduction of 27%. The deterministic model, on the other hand, maintained similar performance on all data, motivating further research into stronger deterministic models.

The most pressing issues for ESA also regard datasets. Datasets for ESA are scarce – especially for the news domain. In an attempt to improve upon this, a dataset has been generated using real-world articles with a novel technique involving world knowledge and annotation by Distant Supervision (DS). Evaluation of the generated dataset shows that already existing datasets, although out-of-domain, may be used as evaluation baselines – enabling efficient verification of data quality for future datasets.

Through an expansive set of experiments on the two main topics, it has been shown that CR can be used to augment the task of ESA. This was done by entity-centric segmentation of longer texts, using resulting coreference links from a CR model. These findings may aid the analysis of data from media outlets and other sources to a greater extent, by attributing sentiment polarity towards entities with higher accuracy. With a refined implementation, this could lead to a detailed view of how the many entities are represented in mass media today. Finally, official sources for datasets have shown interest in the applications for the unified format presented in this thesis, and integration will be initiated in the near future.

9.2. Contributions

There are a few contributions worth mentioning, expanding upon the brief list found in the introduction.

A Unified Format

The defined CorefLite format was largely based on previous work by Lee et al. (2018), converting the OntoNotes dataset from the rather complex CoNLL format to a much more convenient JSON-like format. The CorefLite took this one step further, reducing the format to the common denominator between all available datasets – words and coreference clusters. As shown in this thesis, the three datasets GUM, LitBank and PreCo have been successfully converted to this format, and the official sources for the GUM dataset (Zeldes, 2017) have accepted the format for future integration[1]. At this time, the two other sources for datasets have yet to respond. The conversion system can be found on GitHub[2].

Comprehensive Evaluations

Based on the literature review conducted, as well as all other publications read throughout this thesis, a set of CR models have been extensively tested on a larger selection of datasets from varying domains – for the very first time – with help from the CorefLite formatted datasets. This allows for future researchers to gain detailed insights into the models they may be aiming to develop, as well as for research into the creation and updating of datasets for the task. The commonly used OntoNotes dataset has shown its weaknesses, especially when models are trained on OntoNotes and tested on out-of-domain text. These findings motivate the integration of the CorefLite format as input data for future models to be trained on.

Definition of a Good Coreference Model

Closely related to the previous contribution, a well-generalized CR model has been established – namely SpanBERT (Joshi et al., 2019a). Current literature also places SpanBERT as one of the best performing models, but these results are merely based on evaluations on the OntoNotes dataset. SpanBERT performs well across all datasets tested in this thesis, and can thus be confidently used as a baseline for coming models.

Augmentation with Coreference Resolution

Although the final results are not the most definite, there has been found a positive impact by handling previously unseen data using CR, followed by segmenting the input text on detected entities. This process somewhat alleviates the need for handling long-term

[1] https://github.com/amir-zeldes/gum/pull/58
[2] https://github.com/ph10m/CorefLite

dependencies, and benefits downstream tasks like ESA to tackle individual segments better than the original data.

9.3. Future Work

As the final portion of this thesis, a selection of topics to consider for future work are presented. The two first sections are dedicated to Entity-level Sentiment Analysis (ESA), while the rest regards Coreference Resolution (CR).

9.3.1. Handling Multiple Targets with Attention

The current ill-defined strategy for ESA is worrying from a research perspective. Current work splits the focal point between implicit aspects (Zeng et al., 2019) and explicit entities (Li and Lu, 2019), while both rely on the very same data. This is an undeniable hindrance for further development. The field has, possibly for this reason, no agreed upon methodology as of yet – which was the very basis for attempting to approach it from another angle, through the augmentation of data. For future research into the field – with or without augmentation – there is a merit to employing attention mechanisms (Vaswani et al., 2017) at a much broader level than previously done. Attention mechanisms ultimately hold information for each token in the input data, and can thus handle multiple entities at once with a proper modeling scheme. This simplifies the current approach, which processes the same text numerous times, once for each occurring entity. Regardless of the approach, the current systems are considered to be both ineffective and inefficient. Hopefully, newer research into attention and specialized models for extracting targeted sentiment can resolve this current diversion happening for ESA research.

9.3.2. Metrics for Out-of-domain Evaluation of Sentiment Analysis

For further development of out-of-domain datasets, evaluating on existing datasets has proved a worthy approach. Thus, evaluating with all available data – both training and testing sets – may be desirable. As results in this thesis show, it is desirable to achieve similar performance on both sets in order to create a generalized model. Thus, by using the differences of the predictions for the train and test sets as a heuristic, a metric may be defined by incorporating their respective scores, penalizing large differences. An example of such a metric could be defined as:

$$\text{metric} = \min_{F1 \in \{train, test\}} -\Big(\big(\log(\Delta_d) + 0.0001 \big) \cdot \max_{F1 \in \{train, test\}} \Big) \quad \Delta_d = |F1_{train} - F1_{test}|$$

The addition of 0.0001 is to circumvent the undefined value of $\log(0)$, thus the score would be the same as the test score for a perfect evaluation. The Δ for the Laptop dataset is small ($\Delta_d = 2.26$), which results in a score of $59.15 - \log(2.26) \cdot 61.41 = 37.40$. The Δ is somewhat bigger for the Restaurant dataset ($\Delta_d = 4.23$). Observe how this affects the defined metric: $58.92 - \log(4.23) \cdot 63.15 = 19.37$. A metric of this kind will

help distinguish the models performing great on *one* set, but worse on another of the same data type – promoting well balanced models.

9.3.3. Defining Coreference Entity Importance with Metrics

For the better part of this thesis, the LEA metric has been used, specifically the F1 score. This was based on the work by Moosavi and Strube (2016), attempting to adjust the scores to how *important* an entity is for a given text. As stated when introducing the metric (p. 27), the importance function may be altered according to the desired outcome. This could unfortunately not be prioritized within the scope of this thesis. The importance function is per now defined as default, using the sum of coreference links for an entity as the importance:

$$importance(e) = |e| \tag{9.1}$$

However, further improvement of the equation could involve knowledge graphs. Much alike the process for creating the dataset with the Strise Knowledge Graph, external knowledge graphs can be used (e.g. WikiData) to populate an entity importance score with its number of found relations within an input text. If an article mentions *Satya Nadella* (the Microsoft CEO) once, and then continues mentioning Microsoft ten times, the importance score for Satya Nadella would still be 1. If the importance score is updated to include relations, such as the sum of all its found relations, his assigned importance score would be 11 – which better resembles how a human would intuitively attribute importance (although somewhat naively). A possible implementation is given in Equation 9.2.

$$importance(e) = |e| + |mention(r)\ \forall r \in \mathcal{R}_e| \quad \mathcal{R}_e = \text{relations for entity e} \tag{9.2}$$

9.3.4. An Unconstrained Solution

The already developed system can be slightly modified to support general augmentation, rather than entity-constrained, by reworking the segmentation algorithm. Future systems – although probably not research-focused systems – may benefit from performing a wide analysis of the input data to the model, creating segments for any entity found, as opposed to one single predefined entity. Instead of looking for antecedents for the predefined target, all targets can be passed as input, creating unique segments in an iterative manner. The defined parameters may be modified to avoid a large quantity of segments, however, as some initial examples produced up to hundreds of segments for a single article. Whether this is desired or not must be tested further.

9.3.5. Rectifying Coreference Links with Gradient Boosting

While a good CR model may detect a set of clusters, it cannot always decide between two similar candidates. An intriguing approach was discovered while researching the GAP dataset (Webster et al., 2018), by generating features for the input coreference

data, using a gradient boosting classifier to determine which coreference link is correct for a span of text – provided two or more links. Using textual features, such as those mentioned earlier in Related Work (from p. 35), a gradient boosting approach may help in resolving difficult coreference clusters. A brief experiment on the GAP dataset was conducted using the CatBoost library (Dorogush et al., 2018), achieving higher scores in 30 seconds of training than a selection of hardware-intensive CR models (Clark and Manning, 2015; Wiseman et al., 2016; Lee et al., 2017). Later results fine-tuned BERT models to improve the task (Attree, 2019; Ionita et al., 2019). The results are shown in Table 9.1. The downside of employing a BERT model to complete this task is the computational time required. This process is enabled *after* the initial coreference links have been detected. By adding several BERT models in an ensemble, the system quickly becomes too slow to be useful in real-time applications. Regardless, for future systems (avoiding the ensemble of models), the technique of deciding between two coreference link candidates can be implemented before the final prediction step of a CR system, possibly providing additional robustness.

Model	**Overall F1 Score**
Clark and Manning (2015)	55.0
Wiseman et al. (2016)	64.2
Lee et al. (2017)	64.7
CatBoost	74.7
Ionita et al. (2019)	90.0
Attree (2019)	91.1

Table 9.1.: A selection of models versus the gradient boosting approach using CatBoost on the GAP dataset.

9.3.6. A New, Simpler, Rule-based Model

To further test if rule-based models can perform similarly to deep learning models when exposed to unseen, out-of-domain data, a broader selection of rule-based models should be set up. Results in this thesis show that the rule-based implementation has the least variation of predictions between the evaluated domains of data, but the results were still not optimal. Some suggestions could be to base the model on the features discussed in Hobbs (1978), Lappin and Leass (1994), Lee et al. (2013), Durrett and Klein (2013), Clark and Manning (2015) and Clark and Manning (2016b).

9.3.7. Reworking Models to Train on CorefLite Data

Current models do not support training using the CorefLite formatted data. The models used may be reworked to support the format by reducing the information used in datasets, or by appending empty data fields to the files, in order to mimic the currently used datasets. The former suggestion is preferred. For instance, SpanBERT uses all the

metadata available in the OntoNotes dataset. By reducing the functions to only consider tokens and clusters, the model can be trained on the three other converted datasets in this thesis. An example of metadata used is *speakers* and *genre*. For the interested reader, wanting to take on this work (for SpanBERT specifically), it would involve modifying the *independent.py* file, tracing the changes from the function *get_predictions_and_loss*[3].

9.3.8. Knowledge Graphs and World Knowledge

The incorporation of world knowledge to improve CR has been studied in-depth by Zhang et al. (2019), and deserves more attention in future research. A similar model to that of Zhang et al. (2019), altered with work done on relation extraction as described in Trisedya et al. (2019), serve as an intriguing approach for future models on this topic. As for existing models, NeuralCoref supports the addition of conversion dictionaries (conv_dict for short), allowing to feed the model with knowledge, clarifying information on previously unknown entities. For instance, by adding {"ABC": "company"} gives the model the ability to link references such as "the company" to the entity "ABC". This could additionally be incorporated in a rule-based model, as a custom conversion dictionary can be populated by simply doing look-ups in a knowledge graph. This knowledge-based information can be utilized as a heuristic in determining between coreference links, in a similar manner to how the gradient boosting technique does the same with textual features.

9.3.9. Specification of References in Datasets

As noted by Sukthanker et al. (2018), CR datasets tend to miss certain specifications of what type of resolution they aim at by using a given dataset. Certain references are rare and do not necessarily make sense to be used in the training process of some models. If the desired system will never see the references that are contained within the training dataset, unused references would lead to noise in the data, and possibly cause worse predictions. Therefore, by specifying types of resolution, better models may be developed for specific purposes. Two lists, added in Appendix I.1 and I.2, are composed of constraints and interpretation rules to be considering when handling reference types such as indefinite noun phrases (NPs), definite NPs, demonstratives and several cases of anaphora and cataphora. This may be helpful for future creation of datasets, and the specification of references.

9.3.10. Cross-lingual Coreference Resolution

Ferreira Cruz et al. (2020) discusses cross-lingual aspects of coreference resolution. The findings motivate the following requirements to set up a cross-lingual model:

- A BERT model for a specific language. A Norwegian model was released in March, 2020[4]

[3] https://github.com/mandarjoshi90/coref/blob/master/independent.py#L249
[4] https://github.com/botxo/nordic_bert

- Computing power to fine-tune this with span prediction, e.g. SpanBERT (Joshi et al., 2019a)
- Access to a language-specific generalized language model like those provided by spaCy[5] (Honnibal and Montani, 2017)
- Identifier models to detect part-of-speech (POS) tags, entities and additional textual information. Many of these tasks can also be performed by spaCy

Although this helps many of the preliminary tasks in order to create a CR model, the corpora available for training is still required. A possible suggestion is to use word vectors for a specific language, and thus align the word vectors with similar words from another language (e.g. English). This is feasible as the representation of words are very similar across languages. This has been studied in-depth by Schuster et al. (2019).

9.3.11. Cross-event Coreference Resolution

As opposed to handling coreference in a local context, as has been the case for this thesis, the inclusion of cross-event CR requires the identification of references to certain events. For instance, take the presidential election in 2016. References to this event can be mentioned in newer texts. By creating a system to efficiently hold information across events, the entities "Trump" and "Hillary" can be referenced to the event itself, and thus referenced to texts mentioning the very same event, giving a much richer representation of entities in a global context.

[5] https://spacy.io/usage/models

Bibliography

Oshin Agarwal, Sanjay Subramanian, Ani Nenkova, and Dan Roth. Evaluation of named entity coreference. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–7, Minneapolis, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2801. URL https://www.aclweb.org/anthology/W19-2801.

Dan Ambrošic and Stjepan Dugonjic. Character identification on multiparty dialogues using mention-pair coreference resolution. *Text Analysis and Retrieval 2018 – Course Project Reports. Faculty of Electrical Engineering and Computing – University of Zagreb*, pages 1–5. URL https://www.fer.unizg.hr/_download/repository/TAR-2018-ProjectReports.pdf.

Chinatsu Aone and Scott William Bennett. Evaluating automated and manual acquisition of anaphora resolution strategies. In *33rd Annual Meeting of the Association for Computational Linguistics*, ACL '95, page 122–129, USA, 1995. Association for Computational Linguistics. URL https://doi.org/10.3115/981658.981675.

Rahul Aralikatte, Matthew Lamm, Daniel Hardt, and Anders Søgaard. A simple transfer learning baseline for ellipsis resolution, 2019. URL https://arxiv.org/abs/1908.11141. University of Copenhagen, Stanford University, Copenhagen Business School.

Sandeep Attree. Gendered ambiguous pronouns shared task: Boosting model confidence by evidence pooling. 2019. URL https://arxiv.org/abs/1906.00839.

Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains: A case study. 2005. URL https://www.microsoft.com/en-us/research/publication/customizing-sentiment-classifiers-to-new-domains-a-case-study/.

Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, volume 1, pages 563–566. Granada, International Computer Science Institute – University of California, Berkeley, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.5848.

Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley FrameNet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998. URL https://www.aclweb.org/anthology/P98-1013/.

Breck Baldwin. CogNIAC: high precision coreference with limited knowledge and linguistic resources. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, 1997. URL https://www.aclweb.org/anthology/W97-1306.

David Bamman, Olivia Lewke, and Anya Mansoor. An annotated dataset of coreference in English literature. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 44–54, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.6.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. Revisiting joint modeling of cross-document entity and event coreference resolution. Computer Science Department – Bar-Ilan University, Intel AI Lab – Israel, Ubiquitous Knowledge Processing Lab – Technische Universitat Darmstadt, Germany, 2019. URL https://arxiv.org/abs/1906.01753.

Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Department of Computer Science – University of Illinois, 2008. URL https://www.aclweb.org/anthology/D08-1031.pdf.

Anders Björkelund and Richárd Farkas. Data-driven multilingual coreference resolution using resolver stacking. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 49–55, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W12-4503.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130401. URL https://doi.org/10.1145/130385.130401.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. Google, University of Edinburgh, Cantab Research Ltd – St Johns Innovation Centre, Cambridge, 2013. URL https://arxiv.org/abs/1312.3005.

Henry Yu-Hsin Chen and Jinho D Choi. Character identification on multiparty conversation: Identifying mentions of characters in TV shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, 2016. URL https://www.aclweb.org/anthology/W16-3612/.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

doi: 10.18653/v1/D18-1016. URL https://www.aclweb.org/anthology/D18-1016.

Jinho D. Choi, Joel Tetreault, and Amanda Stent. It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 387–396, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1038. URL https://www.aclweb.org/anthology/P15-1038.

Yejin Choi and Claire Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D08-1083.

Kevin Clark and Christopher D. Manning. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1136. URL https://www.aclweb.org/anthology/P15-1136.

Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas, November 2016a. Association for Computational Linguistics. doi: 10.18653/v1/D16-1245. URL https://www.aclweb.org/anthology/D16-1245.

Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1061. URL https://www.aclweb.org/anthology/P16-1061.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL https://doi.org/10.1023/A:1022627411411.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. SemEval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2089. URL https://www.aclweb.org/anthology/S17-2089.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2009. URL https://www.aclweb.org/anthology/P14-2009.

Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018. URL https://arxiv.org/abs/1810.11363.

Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1203.

Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf.

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1064. URL https://www.aclweb.org/anthology/P19-1064.

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 41–48, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W12-4502.

André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. Coreference resolution: Toward end-to-end and cross-lingual systems. *Information*, 11(2):74, 2020. URL https://www.mdpi.com/2078-2489/11/2/74.

Björn Gambäck, Jussi Karlgren, and Christer Samuelsson. Natural-language interpretation in prolog. 1994. URL https://www.researchgate.net/publication/2827995_Natural-Language_Interpretation_in_Prolog.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2501. URL https://www.aclweb.org/anthology/W18-2501.

Niyu Ge, John Hale, and Eugene Charniak. A statistical approach to anaphora resolution. In *Sixth Workshop on Very Large Corpora*, 1998. URL https://www.aclweb.org/anthology/W98-1119. Dept. of Computer Science, Brown University.

Abbas Ghaddar and Phillippe Langlais. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 136–142, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1021.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ralph Grishman and Beth Sundheim. Message understanding conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL https://www.aclweb.org/anthology/C96-1079.

Jia-Chen Gu, Zhen-Hua Ling, and Nitin Indurkhya. A study on improving end-to-end neural coreference resolution. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, volume 11221, pages 159–169. Springer, ccl 2018, nlp-nabd 2018. lecture notes in computer science edition, 2018. URL https://link.springer.com/chapter/10.1007/978-3-030-01716-3_14.

Liane Guillou, Christian Hardmeier, Aaron Smith, Jörg Tiedemann, and Bonnie Webber. ParCor 1.0: A parallel pronoun-coreference corpus to support statistical MT. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3191–3198, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/298_Paper.pdf.

John M Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. *Unpublished manuscript*, 1971.

Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954. doi: 10.1080/00437956.1954.11659520. URL https://doi.org/10.1080/00437956.1954.11659520.

Olaf Hartig. An initial analysis of facebook's graphql language. Dept. of Computer and Information Science (IDA) – Linköping University, Sweden. Department of Computer Science – Universidad de Chile., 06 2017. URL http://olafhartig.de/files/HartigPerezAMW2017_GraphQL_Preprint.pdf.

Benjamin Heinzerling. *Aspects of Coherence for Entity Analysis*. PhD thesis, Neuphilologische Fakultät > Institut für Computerlinguistik – Heidelberg University, 2019. URL http://www.ub.uni-heidelberg.de/archiv/26117.

Jerry R Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978. URL https://www.sciencedirect.com/science/article/pii/0024384178900062.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017.

C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text, 2014. URL https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109.

Matei Ionita, Yury Kashnitsky, Ken Krige, Vladimir Larin, Atanas Atanasov, and Dennis Logvinenko. Resolving gendered ambiguous pronouns with BERT. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 113–119, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3817. URL https://www.aclweb.org/anthology/W19-3817.

Niklas Jakob and Iryna Gurevych. Using anaphora resolution to improve opinion target identification in movie reviews. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 263–268, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P10-2049.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans, 2019a. URL https://arxiv.org/abs/1907.10529. Allen School of Computer Science Engineering – University of Washington, Seattle, WA. Computer Science Department – Princeton University, Princeton, NJ. Allen Institute of Artificial Intelligence – Seattle. Facebook AI Research, Seattle.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China,

November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1588. URL https://www.aclweb.org/anthology/D19-1588.

Tollef Jørgensen. Multi-Target Entity-Level Sentiment Analysis. Project report in TDT4501, Department of Computer Science, NTNU – Norwegian University of Science and Technology, Dec. 2019.

Ben Kantor and Amir Globerson. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1066. URL https://www.aclweb.org/anthology/P19-1066.

Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

Aliakbar Keshtkaran, Siti Sophiayati Yuhaniz, and Mohammad Reza Rostami. Distributed representation of entity mentions within and across multiple text documents. *Open International Journal of Informatics (OIJI)*, 7(1):35–46, 2019. URL http://apps.razak.utm.my/ojs/index.php/oiji/article/view/195.

Jerzy Krawczuk and Mariusz Ferenc. Coreference resolution for anaphoric pronouns in texts on medical products. *Studies in Logic, Grammar and Rhetoric*, 56(1):205–216, 2018. URL https://content.sciendo.com/view/journals/slgr/56/1/article-p205.xml.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. June 2001.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations, 2019. URL https://arxiv.org/abs/1909.11942. Google Research, Toyota Technological Institute at Chicago.

Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994. URL https://www.aclweb.org/anthology/J94-4002.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1018. URL https://www.aclweb.org/anthology/D17-1018.

Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2108. URL https://www.aclweb.org/anthology/N18-2108.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966. URL https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf. CYBERNETICS AND CONTROL THEORY.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press, 2012. ISBN 9781577355601. URL https://dl.acm.org/doi/10.5555/3031843.3031909.

Hao Li and Wei Lu. Learning latent sentiment scopes for entity-level sentiment analysis, 2017. URL https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14931.

Hao Li and Wei Lu. Learning explicit and implicit structures for targeted sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5478–5488, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1550. URL https://www.aclweb.org/anthology/D19-1550.

Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012. doi: 10.2200/S00416ED1V01Y201204HLT016. URL https://doi.org/10.2200/S00416ED1V01Y201204HLT016.

Bing Liu. *Many Facets of Sentiment Analysis*, pages 11–39. 04 2017. ISBN 978-3-319-55392-4. doi: 10.1007/978-3-319-55394-8_2.

Fei Liu, Luke Zettlemoyer, and Jacob Eisenstein. The referential reader: A recurrent entity network for anaphora resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5925, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1593. URL https://www.aclweb.org/anthology/P19-1593.

Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. Content attention model for aspect based sentiment analysis. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1023–1032, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3186001. URL https://doi.org/10.1145/3178876.3186001.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118117. URL https://doi.org/10.3115/1118108.1118117.

Jing Lu and Vincent Ng. Event coreference resolution: A survey of two decades of research. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5479–5486. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/773. URL https://doi.org/10.24963/ijcai.2018/773.

Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/H05-1004.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014a. Association for Computational Linguistics. doi: 10.3115/v1/P14-5010. URL https://www.aclweb.org/anthology/P14-5010.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014b. URL http://www.aclweb.org/anthology/P/P14/P14-5010.

Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 14(9), 2011. URL https://www.semanticscholar.org/paper/pandas%3A-a-Foundational-Python-Library-for-Data-and-McKinney/1a62eb61b2663f8135347171e30cb9dc0a8931b5.

Andrei Mikheev. Document centered approach to text normalization. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 136–143, New York, NY, USA, 2000. Association

for Computing Machinery. ISBN 1581132263. doi: 10.1145/345508.345564. URL https://doi.org/10.1145/345508.345564.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/2999792.2999959.

Karo Moilanen and Stephen Pulman. Multi-entity sentiment scoring. In *Proceedings of the International Conference RANLP-2009*, pages 258–263, Borovets, Bulgaria, September 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/R09-1048.

Nafise Sadat Moosavi. *Robustness in Coreference Resolution*. PhD thesis, Neuphilologische Fakultät > Institut für Computerlinguistik – Heidelberg University, Heidelberg, 2020. URL https://archiv.ub.uni-heidelberg.de/volltextserver/27919/.

Nafise Sadat Moosavi and Michael Strube. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1060. URL https://www.aclweb.org/anthology/P16-1060.

Nafise Sadat Moosavi and Michael Strube. Lexical features in coreference resolution: To be used with caution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2003. URL https://www.aclweb.org/anthology/P17-2003.

Nafise Sadat Moosavi and Michael Strube. Using linguistic features to improve the generalization capability of neural coreference resolvers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1018. URL https://www.aclweb.org/anthology/D18-1018.

Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073102. URL https://www.aclweb.org/anthology/P02-1014.

Nicolas Nicolov, Franco Salvetti, and Steliana Ivanova. Sentiment analysis: Does coreference matter. In *AISB 2008 convention communication, interaction and social*

intelligence, volume 1, page 37, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.189.3759&rep=rep1&type=pdf.

Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018. URL https://arxiv.org/abs/1811.03378.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Jiaxin Pei, Aixin Sun, and Chenliang Li. Targeted sentiment analysis: A data-driven categorization. In *Proceedings of Sentiment Analysis, Sentiment Analysis, 2019*, 2019. URL https://arxiv.org/abs/1905.03423.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Julien Plu, Roman Prokofyev, Alberto Tonon, Philippe Cudré-Mauroux, Djellel Eddine Difallah, Raphaël Troncy, and Giuseppe Rizzo. Sanaphor++: Combining deep neural networks with semantics for coreference resolution. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1063.

Massimo Poesio and Ron Artstein. Anaphoric annotation in the ARRAU corpus. 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/297_paper.pdf.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. Anaphora resolution with the ARRAU corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 11–22, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-0702. URL https://www.aclweb.org/anthology/W18-0702.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2004. URL https://www.aclweb.org/anthology/S14-2004.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W11-1901.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W12-4501.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. URL https://arxiv.org/abs/2003.07082.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Google, 2019. URL https://arxiv.org/abs/1910.10683.

Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*,

pages 968–977, Singapore, August 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D09-1101.

Altaf Rahman and Vincent Ng. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 814–824, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1082.

Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108, Lisboa, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-2914. URL https://www.aclweb.org/anthology/W15-2914.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4933–4941, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.607.

Craige Roberts. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12(6):683–721, 1989. ISSN 01650157, 15730549. URL http://www.jstor.org/stable/25001367.

Gabi Rolih. Applying coreference resolution for usage in dialog systems. Master's thesis, Uppsala University, Department of Linguistics and Philology, 2018. URL https://uu.diva-portal.org/smash/get/diva2:1218916/FULLTEXT01.pdf.

Frank Rosenblatt. *The perceptron: a theory of statistical separability in cognitive systems (Project Para)*. Cornell Aeronautical Laboratory, 1958. Washington: U.S. Dept. of Commerce, Office of Technical Services.

Sara Rosenthal, Noura Farra, and Preslav Nakov. SemEval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2088. URL https://www.aclweb.org/anthology/S17-2088.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. SentiHood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1146.

Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of twitter. *Inf. Process. Manage.*, 52(1):5–19, January 2016. ISSN 0306-4573. doi: 10.1016/j.ipm.2015.01.005. URL https://doi.org/10.1016/j.ipm.2015.01.005.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *EMC^2: 5th Edition Co-located with NeurIPS'19*, 2019. URL https://arxiv.org/abs/1910.01108.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1162. URL https://www.aclweb.org/anthology/N19-1162.

DAVID W. SCOTT. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 12 1979. ISSN 0006-3444. doi: 10.1093/biomet/66.3.605. URL https://doi.org/10.1093/biomet/66.3.605.

Tomohide Shibata and Sadao Kurohashi. Entity-centric joint modeling of Japanese coreference resolution and predicate argument structure analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 579–589, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1054. URL https://www.aclweb.org/anthology/P18-1054.

Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure, 2019. URL https://arxiv.org/abs/1912.05848.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. Targeted sentiment classification with attentional encoder network. *Lecture Notes in Computer Science*, page 93–103, 2019. ISSN 1611-3349. doi: 10.1007/978-3-030-30490-4_9. URL http://dx.doi.org/10.1007/978-3-030-30490-4_9.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27

(4):521–544, 2001. doi: 10.1162/089120101753342653. URL https://www.aclweb.org/anthology/J01-4004.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, pages 11–21, 1972.

Stian Steinbakken. Paying attention to native-language identification. MSc Thesis, Dept. of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway, 2019.

Nikolaos Stylianou and Ioannis Vlahavas. A neural entity coreference resolution review, 2019. URL https://arxiv.org/abs/1910.09329. School of Informatics – Aristotle University of Thessaloniki.

Sanjay Subramanian and Dan Roth. Improving generalization in coreference resolution via adversarial training. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 192–197, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-1021. URL https://www.aclweb.org/anthology/S19-1021.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936547. doi: 10.1145/1242572.1242667. URL https://doi.org/10.1145/1242572.1242667.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. Anaphora and coreference resolution: A review. *Information Fusion*, 59, 2018. URL https://www.sciencedirect.com/science/article/pii/S1566253519303677.

Nicholas Tan and Hongshen Zhao. Gender balanced coreference resolution. *32nd Conference on Neural Information Processing Systems (NIPS 2018), Montral, Canada*, 2019. URL https://www.semanticscholar.org/paper/Gender-Balanced-Coreference-Resolution-Tan/bc02933b8cf7745370881559a64da3cc871fe61d. Stanford University.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1311.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1023. URL https://www.aclweb.org/anthology/P19-1023.

Olga Uryupina, Massimo Poesio, Claudio Giuliano, and Kateryna Tymoshenko. Disambiguation and filtering methods in using web knowledge for coreference resolution. 01 2011. doi: 10.4018/978-1-61350-447-5.ch013. URL https://www.researchgate.net/publication/221439091_Disambiguation_and_Filtering_Methods_in_Using_Web_Knowledge_for_Coreference_Resolution.

Rob A Van der Sandt. Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, 9(4):333–377, 11 1992. ISSN 0167-5133. doi: 10.1093/jos/9.4.333. URL https://doi.org/10.1093/jos/9.4.333.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Devendra Kumar Verma and Pushpak Bhattacharyya. Survey on coreference resolution, relation and event extraction. 2018. URL https://www.semanticscholar.org/paper/Survey-on-Coreference-Resolution-%2C-Relation-and-Verma-Bhattacharyya/42990bc6645f53c89ff3a7e8ec98d1cb93fec58d. Dept. of Computer Science and Engineering –Indian Institute of Technology, Bombay.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. URL https://www.aclweb.org/anthology/M95-1005.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Hongning Wang and ChengXiang Zhai. Generative models for sentiment analysis and opinion mining. In *A practical guide to sentiment analysis*, pages 107–134. Springer, 2017.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617, 2018. doi: 10.1162/tacl_a_00240. URL https://www.aclweb.org/anthology/Q18-1042.

Terry Winograd. *Understanding Natural Language*. Academic Press, Inc., USA, 1972. ISBN 0127597506.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the*

53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1416–1426, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1137. URL https://www.aclweb.org/anthology/P15-1137.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1114. URL https://www.aclweb.org/anthology/N16-1114.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. URL https://arxiv.org/abs/1910.03771. HuggingFace.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. Coreference resolution as query-based span prediction. *ShannonAI, Stanford University*, 2019. URL https://arxiv.org/abs/1911.01746.

Heng Yang, Biqing Zeng, JianHao Yang, Youwei Song, and Ruyang Xu. A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction, 2019a. URL https://arxiv.org/abs/1912.07976. School of Computer, South China Normal University, Guangzhou 510631, China. School of Software, South China Normal University, Foshan 528225, China. Baidu Inc., Beijing 100085, China.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019b. Carnegie Mellon University, Google AI Brain Team.

Amir Zeldes. The gum corpus: Creating multilayer resources in the classroom. *Lang. Resour. Eval.*, 51(3):581–612, September 2017. ISSN 1574-020X. doi: 10.1007/s10579-016-9343-x. URL https://doi.org/10.1007/s10579-016-9343-x.

Amir Zeldes and Shuo Zhang. When annotation schemes change rules help: A configurable approach to coreference resolution beyond OntoNotes. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, pages 92–101, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-0713. URL https://www.aclweb.org/anthology/W16-0713.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16): 3389, 2019. URL https://www.mdpi.com/2076-3417/9/16/3389.

Hongming Zhang, Yan Song, and Yangqiu Song. Incorporating context and external knowledge for pronoun coreference resolution. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 872–881, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1093. URL https://www.aclweb.org/anthology/N19-1093.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2017. URL https://www.aclweb.org/anthology/P18-2017.

Appendices

A. Literature Review Tables

In the following subsections are the tables referred to in the Literature Review (Section 3.1).

A.1. Query Q1

Initially gathered material using query Q1 is found in Table A.1, where the result of the quality assessment can be found in Table A.2

A.2. Query Q2

Initially gathered material using query Q2 is found in Table A.3, where the result of the quality assessment can be found in Table A.4.

A.3. Final Review Library

The final review library is shown in Table A.5.

B. Sentiment Dataset Analysis

Figures B.1, B.2, B.3, B.4 and B.5 contain a more detailed view of the document length distribution discussed in Section 4.2.

ID	Title	Author(s)
Q1_01	Sanaphor++: Combining Deep Neural Networks with Semantics for Coreference Resolution	Plu et al. (2018)
Q1_02	Revisiting Joint Modeling of Cross-document Entity and Event Coreference Resolution	Barhom et al. (2019)
Q1_03	SpanBERT: Improving Pre-training by Representing and Predicting Spans	Joshi et al. (2019a)
Q1_04	Evaluation of Named Entity Coreference	Agarwal et al. (2019)
Q1_05	Character Identification on Multiparty Dialogues Using Mention-Pair Coreference Resolution	Ambrošic and Dugonjic
Q1_06	Neural Relation Extraction for Knowledge Base Enrichment	Trisedya et al. (2019)
Q1_07	Aspects of Coherence for Entity Analysis	Heinzerling (2019)
Q1_08	Incorporating Context and External Knowledge for Pronoun Coreference Resolution	Zhang et al. (2019)
Q1_09	Coreference Resolution: Toward End-to-End and Cross-Lingual Systems	Ferreira Cruz et al. (2020)
Q1_10	Survey on Coreference Resolution, Relation and Event Extraction	Verma and Bhattacharyya (2018)
Q1_11	Applying Coreference Resolution for Usage in Dialog Systems	Rolih (2018)
Q1_12	Distributed Representation of Entity Mentions Within and Across Multiple Text Documents	Keshtkaran et al. (2019)
Q1_13	Event Coreference Resolution: A Survey of Two Decades of Research	Lu and Ng (2018)

Table A.1.: Retrieved publications for query $Q1$

ID	QC1	QC2	QC3	QC4	QC5	QC6	Score
Q1_01	1	0.5	0	0	1	1	3.5
Q1_02	0.5	1	0	1	0	0.5	3
Q1_03	1	1	1	1	0	1	5
Q1_04	1	1	1	0	0	1	4
Q1_05	0.5	0.5	0.5	1	0	0	2.5
Q1_06	0.5	1	0.5	1	1	0.5	4.5
Q1_07	0.5	0	0.5	0	0	0	1
Q1_08	1	1	1	1	1	1	6
Q1_09	1	1	1	0	1	1	5
Q1_10	1	0	0	N/A	N/A	N/A	1
Q1_11	1	0	0	0	0	0	1
Q1_12	1	0	0	0	0	0	1
Q1_13	1	1	1	0	1	0	4

Table A.2.: Results for query $Q1$. Publications with a quality assessment score ≥ 4 marked as green.

ID	Title	Author(s)
Q2_01	Higher-order Coreference Resolution with Coarse-to-fine Inference	Lee et al. (2018)
Q2_02	Anaphora and Coreference Resolution: A Review	Sukthanker et al. (2018)
Q2_03	Entity-Centric Joint Modeling of Japanese Coreference Resolution and Predicate Argument Structure Analysis	Shibata and Kurohashi (2018)
Q2_04	Neural Coreference Resolution with Deep Biaffine Attention by Joint Mention Detection and Mention Clustering	Zhang et al. (2018)
Q2_05	End-to-end Deep Reinforcement Learning Based Coreference Resolution	Fei et al. (2019)
Q2_06	Coreference Resolution with Entity Equalization	Kantor and Globerson (2019)
Q2_07	BERT for Coreference Resolution: Baselines and Analysis	Joshi et al. (2019b)
Q2_08	Gender Balanced Coreference Resolution	Tan and Zhao (2019)
Q2_09	Coreference Resolution as Query-based Span Prediction	Wu et al. (2019)
Q2_10	Coreference Resolution for Anaphoric Pronouns in Texts on Medical Products	Krawczuk and Ferenc (2018)
Q2_11	A Study on Improving End-to-End Neural Coreference Resolution	Gu et al. (2018)
Q2_12	Anaphora resolution with the ARRAU corpus	Poesio et al. (2018)
Q2_13	Robustness in Coreference Resolution	Moosavi (2020)
Q2_14	A Neural Entity Coreference Resolution Review	Stylianou and Vlahavas (2019)
Q2_15	Improving Generalization in Coreference Resolution via Adversarial Training	Subramanian and Roth (2019)
Q2_16	The referential reader: A recurrent entity network for anaphora resolution	Liu et al. (2019)
Q2_17	Ellipsis and Coreference Resolution as Question Answering	Aralikatte et al. (2019)

Table A.3.: Retrieved publications for query $Q2$

ID	QC1	QC2	QC3	QC4	QC5	QC6	Score
Q2_01	1	1	1	1	1	1	6
Q2_02	1	1	1	N/A	N/A	N/A	3+
Q2_03	1	0	0	0	0	0	1
Q2_04	1	1	1	0	0	1	4
Q2_05	1	1	1	0	0	1	4
Q2_06	1	1	0.5	1	0	1	4.5
Q2_07	1	1	1	1	0	0.5	4.5-
Q2_08	1	0.5	0.5	0.5	0	0	2.5
Q2_09	1	1	1	1	0.5	1	5.5
Q2_10	1	0.5	0	0	0	0	1.5
Q2_11	1	1	1	0	0	0.5	3.5
Q2_12	0.5	1	0.5	0	0	1	3
Q2_13	1	1	1	0.5	1	0.5	5
Q2_14	1	1	1	N/A	N/A	N/A	3+
Q2_15	1	1	0.5	1	0.5	0	4
Q2_16	1	1	1	1	0	1	5
Q2_17	1	1	0.5	0	0.5	0	3

Table A.4.: Results for query $Q2$. Publications with a quality assessment score ≥ 4 marked as green. + indicates overridden importance as certain assessments were not applicable (in this case, larger reviews). - indicates an outdated version of an already included publication, thus being superseded by the newest version (ID Q2_07 is superseded by ID Q1_03).

ID	Title	Author(s)	Score
Q1_08	Incorporating Context and External Knowledge for Pronoun Coreference Resolution	Zhang et al. (2019)	6
Q2_01	Higher-order Coreference Resolution with Coarse-to-fine Inference	Lee et al. (2018)	6
Q2_09	Coreference Resolution as Query-based Span Prediction	Wu et al. (2019)	5.5
Q1_03	SpanBERT: Improving Pre-training by Representing and Predicting Spans	Joshi et al. (2019a)	5
Q1_09	Coreference Resolution: Toward End-to-End and Cross-Lingual Systems	Ferreira Cruz et al. (2020)	5
Q2_13	Robustness in Coreference Resolution	Moosavi (2020)	5
Q2_16	The referential reader: A recurrent entity network for anaphora resolution	Liu et al. (2019)	5
Q1_06	Neural Relation Extraction for Knowledge Base Enrichment	Trisedya et al. (2019)	4.5
Q2_06	Coreference Resolution with Entity Equalization	Kantor and Globerson (2019)	4.5
Q1_04	Evaluation of Named Entity Coreference	Agarwal et al. (2019)	4
Q1_13	Event Coreference Resolution: A Survey of Two Decades of Research	Lu and Ng (2018)	4
Q2_04	Neural Coreference Resolution with Deep Biaffine Attention by Joint Mention Detection and Mention Clustering	Zhang et al. (2018)	4
Q2_05	End-to-end Deep Reinforcement Learning Based Coreference Resolution	Fei et al. (2019)	4
Q2_15	Improving Generalization in Coreference Resolution via Adversarial Training	Subramanian and Roth (2019)	4
Q2_02	Anaphora and Coreference Resolution: A Review	Sukthanker et al. (2018)	3
Q2_14	A Neural Entity Coreference Resolution Review	Stylianou and Vlahavas (2019)	3

Table A.5.: Final review libary from queries $Q1$ and $Q2$, sorted by quality assessment score.

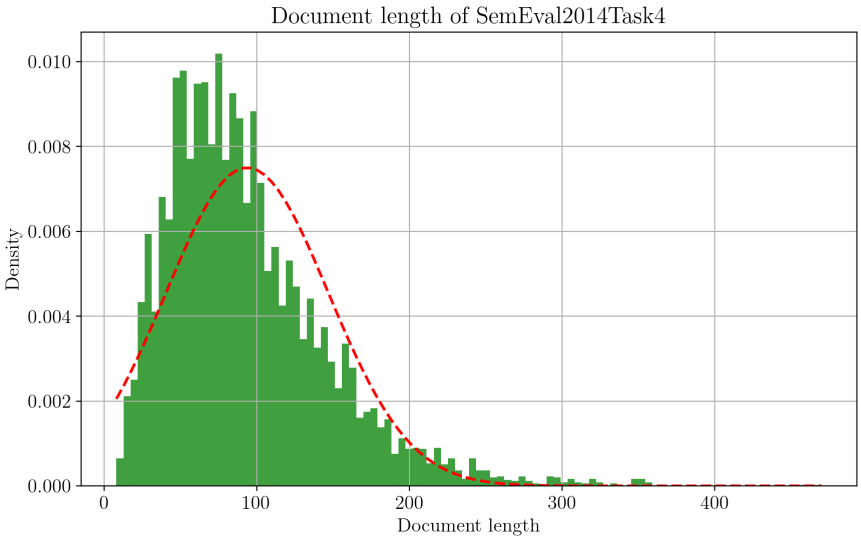


Figure B.1.: Density distribution of document length for SemEval 2014, Task 4

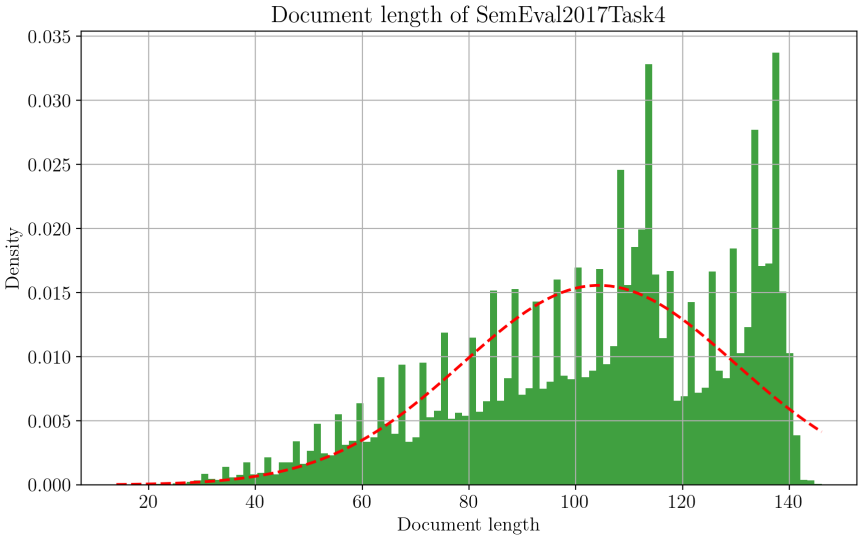


Figure B.2.: Density distribution of document length for SemEval 2017, Task 4

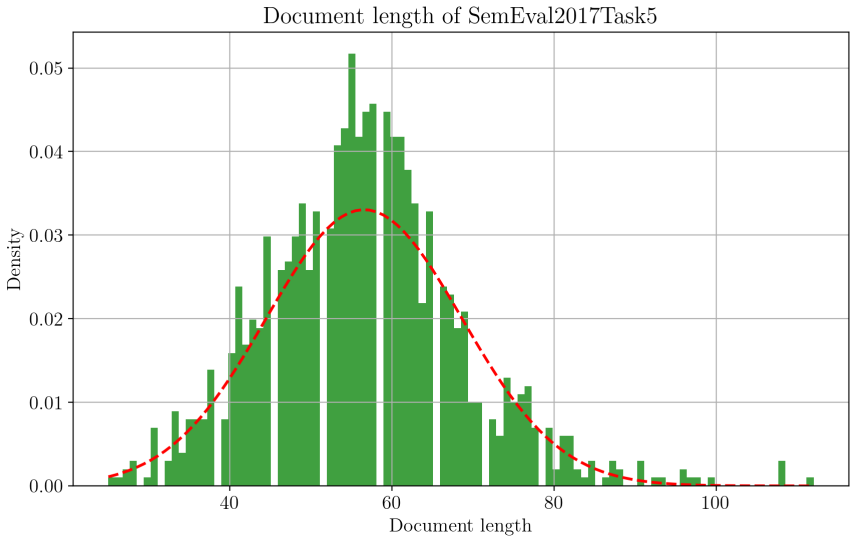


Figure B.3.: Density distribution of document length for SemEval 2017, Task 5

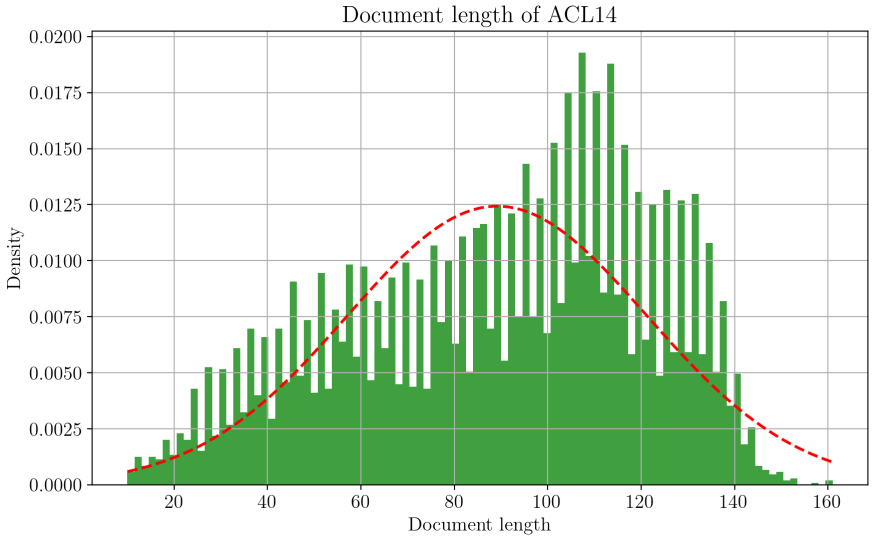


Figure B.4.: Density distribution of document length for ACL-14

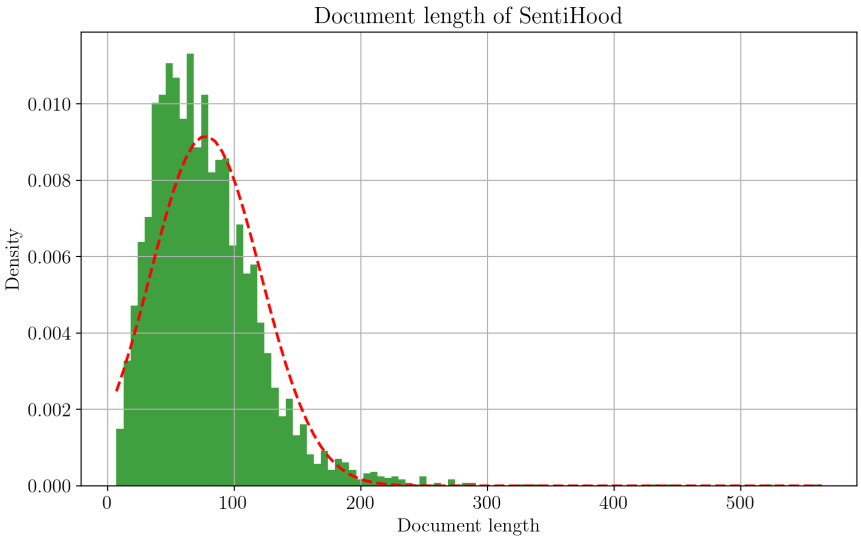


Figure B.5.: Density distribution of document length for SentiHood

C. Coreference Dataset Analysis

The plots in Figures C.1, C.2, C.3 and C.4 illustrate the correlation between document length and coreference links, by the regression line (with a highlighted confidence interval of 95%) in the scatter plots. The curves for document length and coreference links are a result of a kernel density estimation with a Scott estimate (SCOTT, 1979). The results for LitBank (Figure C.3) are unique, as the dataset primarily has longer documents (but of similar length), all with varying number of annotated coreference links.

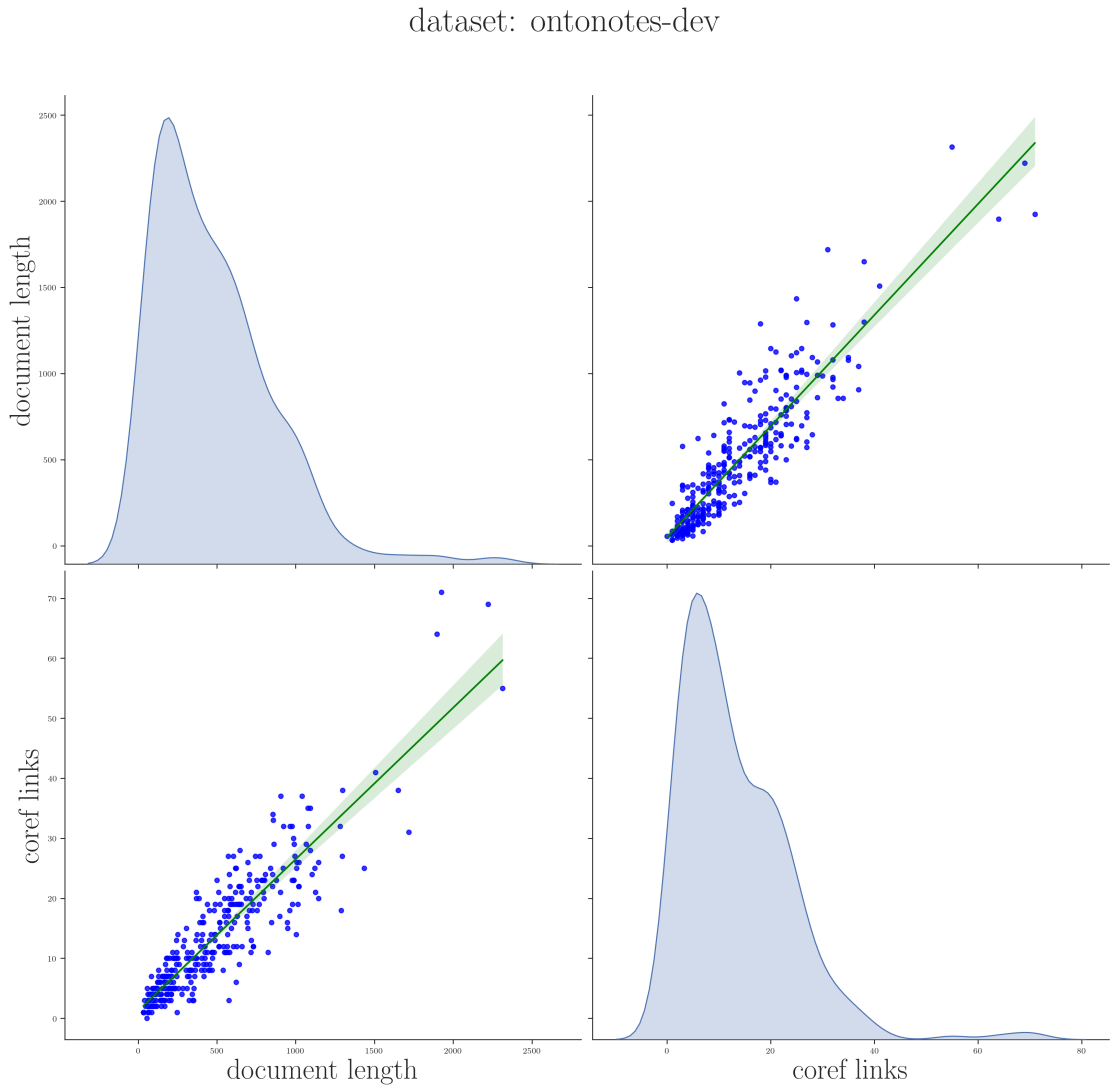


Figure C.1.: Pairwise plot of document length and coreference links for the Ontonotes (dev) dataset.

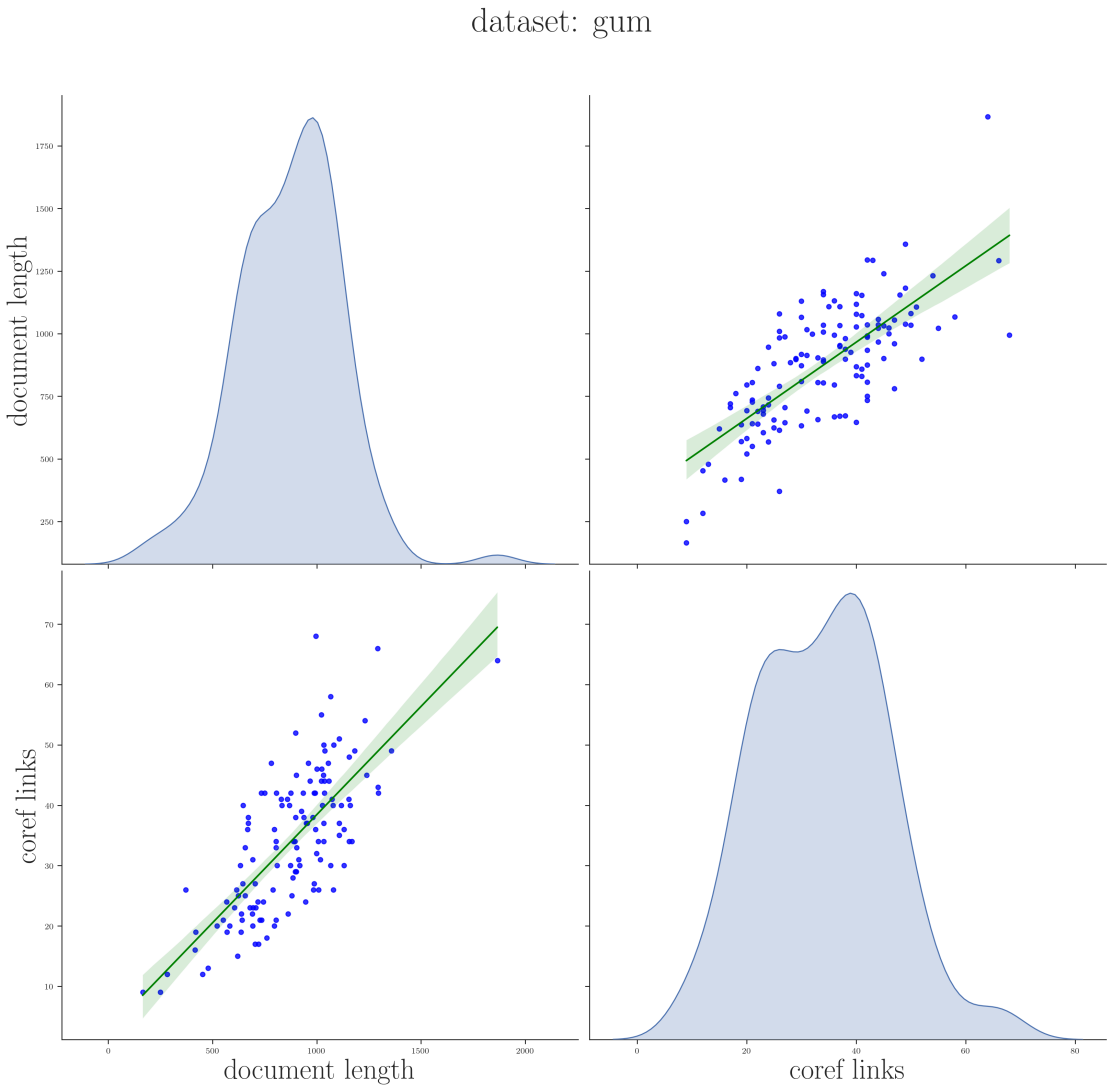


Figure C.2.: Pairwise plot of document length and coreference links for the GUM dataset.

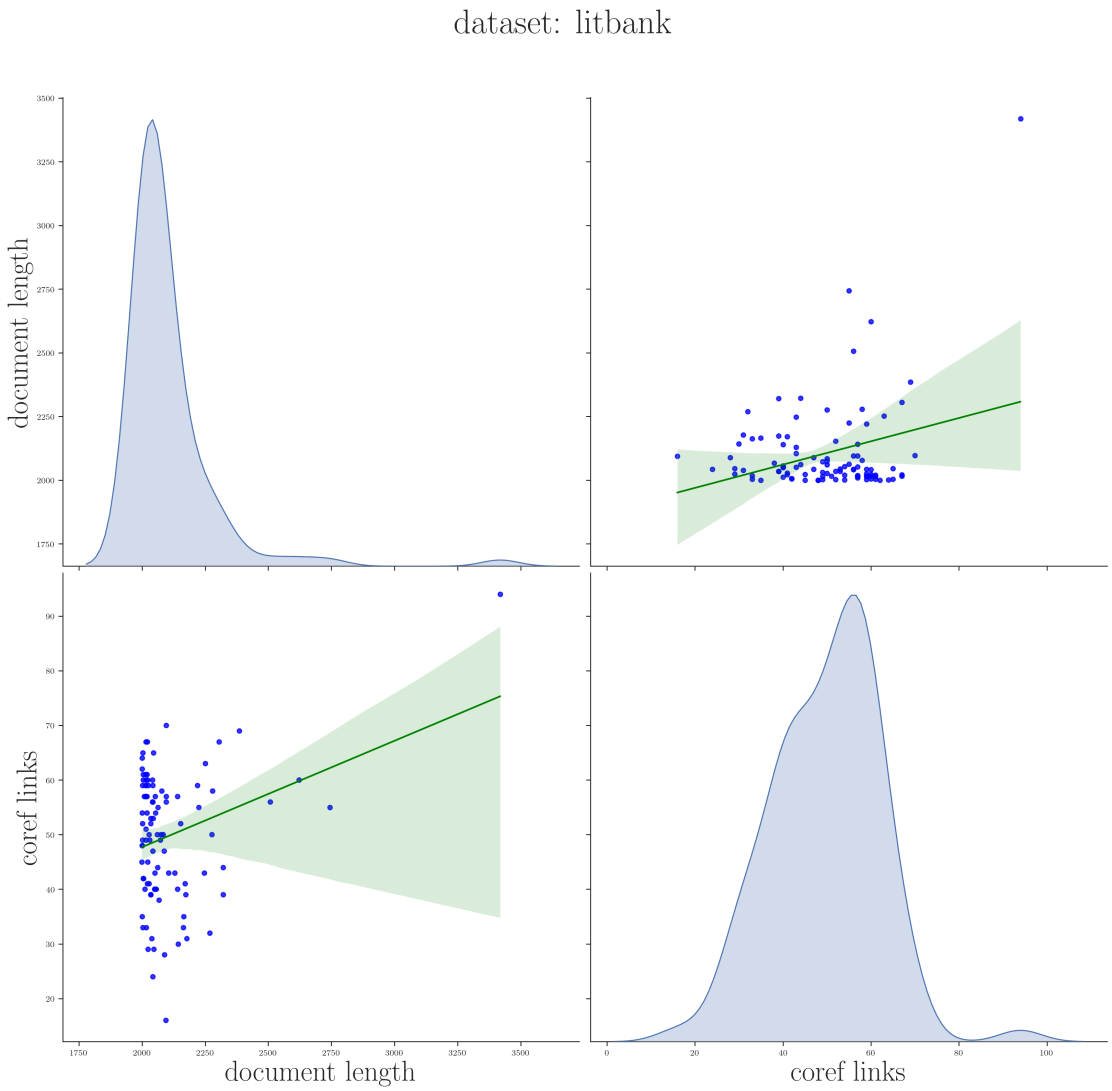


Figure C.3.: Pairwise plot of document length and coreference links for the LitBank dataset. Note that the similarly sized documents (clustering around $n = 2100$) greatly differ in number of coreference links.

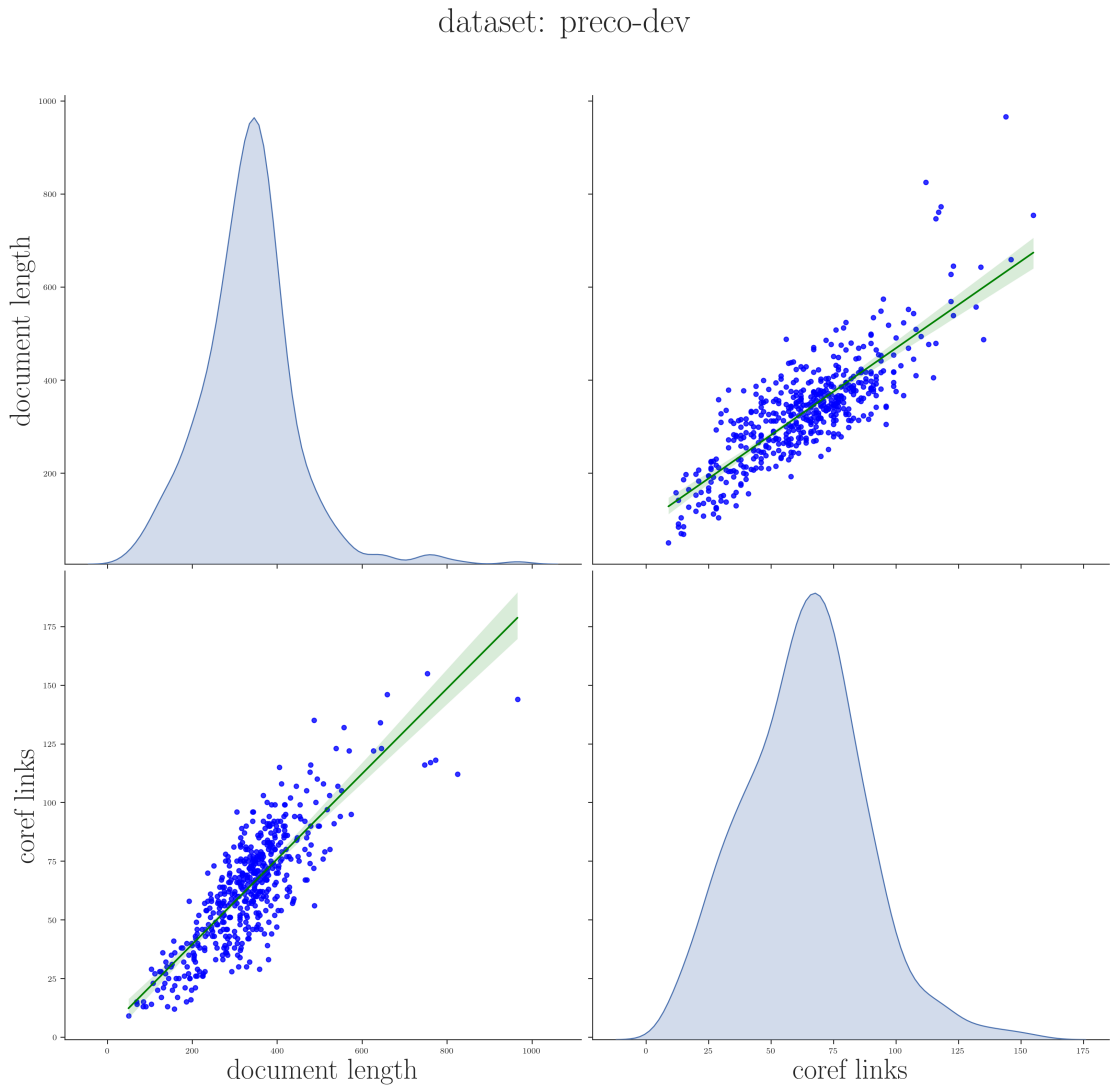


Figure C.4.: Pairwise plot of document length and coreference links for the PreCo (dev) dataset.

D. NeuralCoref

This part of the Appendix considers the formatting and hyperparameters of the NeuralCoref model.

D.1. Numpy Array Formatting

```
parsed_ontonotes/
├── train
│   ├── numpy
│   │   ├── conll_tokens.bin
│   │   ├── doc.bin
│   │   ├── locations.bin
│   │   ├── mentions_features.npy
│   │   ├── mentions_labels.npy
│   │   ├── mentions_pairs_length.npy
│   │   ├── mentions_pairs_start_index.npy
│   │   ├── mentions_spans.npy
│   │   ├── mentions_words.npy
│   │   ├── pairs_ant_index.npy
│   │   ├── pairs_features.npy
│   │   ├── pairs_labels.npy
│   │   ├── spacy_lookup.bin
│   │   ├── static_word_embeddings.npy
│   │   ├── static_word_vocabulary.txt
│   │   ├── tuned_word_embeddings.npy
│   │   └── tuned_word_vocabulary.txt
│   └── train.english.v4_gold_conll
├── dev
│   └── ...
└── test
    └── ...
```

D.2. Hyperparameters

Hyperparameters for NeuralCoref found in Table D.1.

D.3. Testing Greedyness

Effect of changing greedyness on the OntoNotes test set are found in Table D.2.

Parameter	Value	Description
batchsize	20000	Number of mention pairs to be processed in one batch
numworkers	8	Number of workers to load batches
costfn	0.8	Cost of a false new
costfl	0.4	Cost of a false link
costwl	1.0	Cost of a wrong link
h1	1000	Hidden units on first layer
h2	500	Hidden units on second layer
h3	500	Hidden units on third layer
all_pairs_epoch	200	Epochs for mention pairs pre-training
top_pairs_epoch	200	Epochs for top-pairs pre-training
ranking_epoch	200	Epochs for ranking training
all_pairs_lr	2e-4	Mention pairs learning rate
top_pairs_l2	2e-4	Top pairs learning rate
ranking_lr	2e-6	Ranking learning rate
all_pairs_l2	1e-6	Mention pairs l2 regularization
top_pairs_l2	1e-5	Top pairs l2 regularization
ranking_l2	1e-5	Ranking training l2 regularization
patience	3	Epochs before considering decrease in evaluation metric
min_lr	2e-8	Minimum learning rate
on_eval_decrease	next_stage	What to do when evaluation metric decreases
lazy	1	Lazy loading of numpy files

Table D.1.: Hyperparameters for the NeuralCoref Training Process

E. Model Configurations for Coreference Resolution

For e2e-coref and SpanBERT, the configuration files (both named *experiments.conf*) were used as default from their respective official sources, as seen listed below. The files can also be found in provided code. The only changes made were to direct the data sources toward the correct local dataset destination on the IDUN cluster.

- e2e-coref: `https://github.com/kentonl/e2e-coref/blob/master/experiments.conf`
- SpanBERT: `https://github.com/mandarjoshi90/coref/blob/master/experiments.conf`

Additionally, to enable training on the cluster, some kernel files had to be modified to remove unsupported libraries on the cluster. The source file can be found at `https://github.com/mandarjoshi90/coref/blob/master/setup_all.sh`. The last line was changed from

```
g++ -std=c++11 -shared coref_kernels.cc -o coref_kernels.so -
   fPIC ${TF_CFLAGS[@]} ${TF_LFLAGS[@]} -O2 -
   D_GLIBCXX_USE_CXX11_ABI=0
```

Greedyness NeuralCoref Pre-trained	**CoNLL F1 OntoNotes test set**
0.51	0.493389
0.511	0.493349
0.512	0.493149
0.513	0.493460
0.514	0.493759
0.515	0.494050
0.516	0.493991
0.517	0.493720
0.518	0.494006
0.519	0.494415
0.520	0.494371
0.521	0.494301
0.522	0.494890
0.524	0.495060
0.525	0.495241
0.526	0.495170
0.527	0.494885
0.528	0.494863

Table D.2.: NeuralCoref Greedyness Parameter and the respective CoNLL F1 scores on the OntoNotes Test Set

to

```
g++ -std=c++11 -shared coref_kernels.cc -o coref_kernels.so -
   fPIC ${TF_CFLAGS[@]} \${TF_LFLAGS[@]} -O2
```

The training process on the cluster for both e2e-coref and SpanBERT are compared in Figure E.1. The x-axis represents iterations. The e2e-coref was left training for 96 hours (never exits unless manually enforced), and the SpanBERT model ran for approximately 8 hours.

F. Annotation Tool

A screenshot of the developed annotation tool can be seen in Figure F.1. The full system is available at `https://github.com/ph10m/PandAnnotator`.

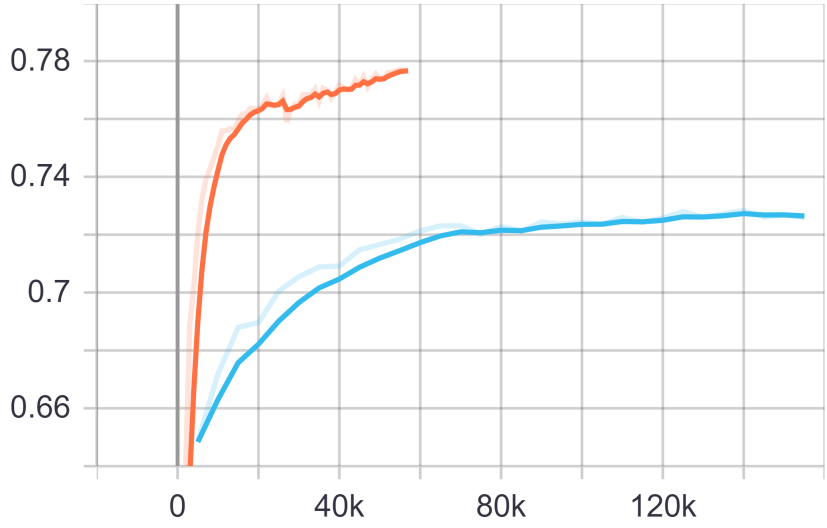


Figure E.1.: SpanBERT (orange line) and e2e-coref (blue line) and iteration steps on the IDUN Cluster.

G. Evaluation Tables

G.1. Out-of-Domain

Full out-of-domain results for OntoNotes, GUM, LitBank and PreCo are in Tables G.1, G.2, G.3 and G.4.

G.2. In-domain

Full evaluation results for OntoNotes (news) are in Table G.5 and GUM (news) in Table G.6.

G.3. Unmodified Datasets

Unmodified dataset evaluations (continuing from Chapter 6) are in Tables G.7 and G.8.

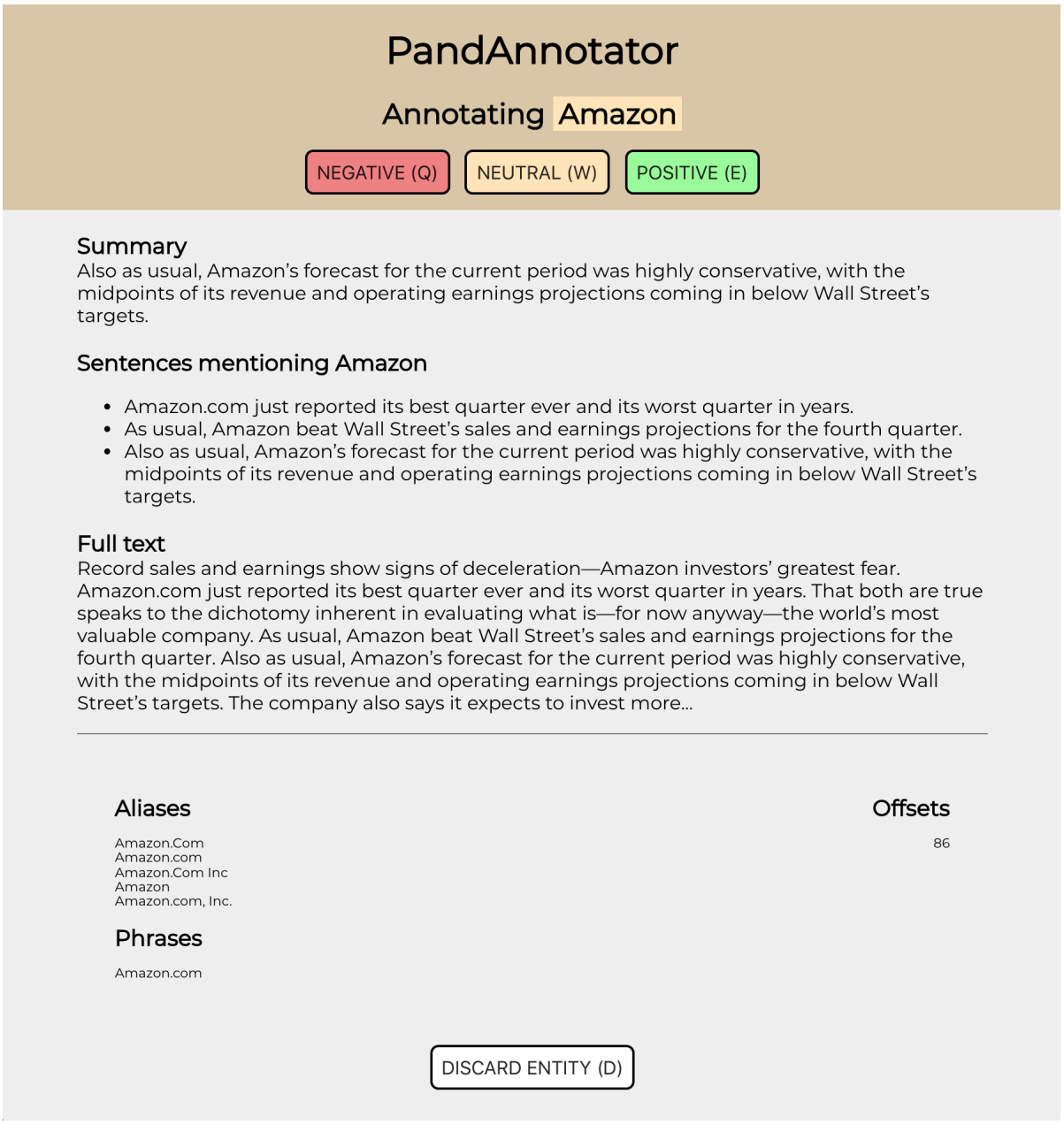


Figure F.1.: The Pandas Dataframe Annotation Tool

Dataset	**MUC**			**B-CUBED**			**CEAF**			**LEA**			**CoNNL**
PreCo	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
CoreNLP Deterministic	57.56	61.48	59.46	47.73	47.01	47.37	38.78	51.32	44.17	42.54	41.28	41.90	50.33
CoreNLP Statistical	71.73	65.85	68.66	59.51	49.58	54.10	51.60	47.20	49.31	55.49	45.36	49.92	57.36
NeuralCoref	73.46	45.63	56.30	65.32	34.97	45.55	55.88	43.90	49.17	60.81	31.52	41.52	50.34
SpanBERT	**83.65**	**83.38**	**83.51**	**74.83**	**74.99**	**74.91**	**74.47**	**73.41**	**73.94**	**72.50**	**72.68**	**72.59**	**77.45**

Table G.1.: Out-of-domain evaluations on the OntoNotes dataset (no news)

Dataset	MUC			B-CUBED			CEAF			LEA			CoNNL
GUM	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
CoreNLP Deterministic	57.42	41.08	47.90	50.19	27.69	35.69	38.46	**33.49**	35.81	43.70	23.02	30.16	39.80
CoreNLP Statistical	76.22	43.33	55.25	67.01	27.21	38.70	47.80	25.41	33.18	62.23	24.07	34.71	42.38
NeuralCoref	73.10	26.77	39.19	67.00	15.59	25.30	45.92	19.26	27.14	61.18	13.19	21.70	30.54
SpanBERT	**79.56**	**46.56**	**58.74**	**72.37**	**35.59**	**47.71**	**59.65**	32.78	**42.31**	**68.40**	**32.87**	**44.41**	**49.59**

Table G.2.: Out-of-domain evaluations on the GUM dataset (no news)

Dataset	**MUC**			**B-CUBED**			**CEAF**			**LEA**			**CoNNL**
LitBank	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
CoreNLP Deterministic	56.08	66.30	60.76	40.89	35.29	37.88	15.51	49.43	23.61	37.84	32.42	34.92	40.75
CoreNLP Statistical	72.54	67.52	69.94	51.80	34.31	41.28	22.52	39.38	28.66	49.36	31.87	38.73	46.63
NeuralCoref	71.18	45.96	55.86	56.23	21.71	31.32	24.60	39.24	30.24	53.14	19.55	28.59	39.14
SpanBERT	**73.29**	**77.92**	**75.53**	**56.93**	**61.63**	**59.19**	**32.91**	**58.24**	**42.06**	**55.41**	**59.92**	**57.58**	**58.92**

Table G.3.: Out-of-domain evaluations on the LitBank dataset

Dataset	**MUC**			**B-CUBED**			**CEAF**			**LEA**			**CoNNL**
PreCo	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
CoreNLP Deterministic	63.72	48.73	55.22	56.67	38.10	45.57	44.93	44.58	44.76	51.12	33.69	40.61	48.52
CoreNLP Statistical	76.14	53.16	62.61	68.09	40.02	50.41	56.01	39.09	46.05	63.93	36.51	46.48	53.02
NeuralCoref	74.24	33.88	46.53	68.50	25.34	37.00	55.82	32.65	41.20	63.43	22.42	33.13	41.57
SpanBERT	**79.47**	**53.70**	**64.09**	**72.99**	**45.05**	**55.71**	**63.80**	**46.54**	**53.82**	**70.05**	**42.48**	**52.89**	**57.87**

Table G.4.: Out-of-domain evaluations on the PreCo dataset

Dataset	**MUC**			**B-CUBED**			**CEAF**			**LEA**			**CoNNL**
PreCo	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
CoreNLP Deterministic	45.60	53.40	49.20	40.31	44.21	42.17	36.82	50.12	42.45	34.71	37.84	36.21	44.61
CoreNLP Statistical	66.68	52.68	58.86	59.31	41.67	48.95	53.57	43.16	47.80	54.47	36.92	44.01	51.87
NeuralCoref	66.06	41.95	51.32	62.11	31.57	41.86	54.66	38.70	45.31	56.74	27.34	36.90	46.16
SpanBERT	**75.12**	**74.71**	**74.91**	**69.42**	**68.89**	**69.15**	**70.91**	**69.81**	**70.35**	**66.39**	**69.81**	**66.02**	**71.47**

Table G.5.: In-domain evaluations on the OntoNotes dataset (news)

Dataset	**MUC**			**B-CUBED**			**CEAF**			**LEA**			**CoNNL**
PreCo	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
CoreNLP Deterministic	55.86	40.88	47.21	50.40	29.93	37.56	40.99	35.60	38.10	44.02	25.16	32.02	40.96
CoreNLP Statistical	69.49	36.48	47.85	62.28	24.54	35.21	49.04	28.58	36.11	56.88	21.03	30.70	39.72
NeuralCoref	62.89	26.22	37.01	57.06	16.35	25.41	43.87	22.79	30.00	50.64	13.41	21.20	30.81
SpanBERT	**72.30**	**47.18**	**57.10**	**64.72**	**38.06**	**47.93**	**57.44**	**37.77**	**45.57**	**60.28**	**34.75**	**44.10**	**50.20**

Table G.6.: In-domain evaluations on the GUM dataset (news)

Dataset OntoNotes	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	57.21	46.18	43.72	49.04	40.69	50.52	40.61
CoreNLP Statistical	66.71	53.02	48.93	56.22	55.29	43.49	48.69
NeuralCoref	55.24	44.74	48.21	49.40	59.91	30.59	40.50
SpanBERT	**81.71**	**73.64**	**73.01**	**76.12**	**71.15**	**71.13**	**71.14**

Table G.7.: F1 evaluations + LEA metric on the full OntoNotes test dataset

Dataset GUM	**MUC**	**B-CUBED**	**CEAF**	**CoNLL**	**LEA**		
	F1	F1	F1	F1	Prec.	Rec.	F1
CoreNLP Deterministic	47.90	35.69	35.81	39.80	43.70	23.02	30.16
CoreNLP Statistical	55.25	38.70	33.18	42.38	62.23	24.07	34.71
NeuralCoref	39.19	25.30	27.14	30.54	61.18	13.19	21.70
SpanBERT	**58.74**	**47.71**	**42.31**	**49.59**	**68.40**	**32.87**	**44.41**

Table G.8.: F1 evaluations + LEA metric on the full GUM dataset

H. DistilBERT SST-2 Configuration

Parameter	**Value**
activation	gelu
architectures	["DistilBertForSequenceClassification"]
attention_dropout	0.1
dim	768
dropout	0.1
finetuning_task	"sst-2"
hidden_dim	3072
id2label	{ "0": "NEGATIVE", "1": "POSITIVE" }
initializer_range	0.02
label2id	{ "NEGATIVE": 0, "POSITIVE": 1 }
max_position_embeddings	512
model_type	"distilbert"
n_heads	12
n_layers	6
output_past	true
pad_token_id	0
qa_dropout	0.1
seq_classif_dropout	0.2
sinusoidal_pos_embds	false
tie_weights_	true
vocab_size	30522

Table H.1.: DistilBERT fine-tuning configuration

I. Future Work – Rule-based Models

The two appendices found here contain some constraints and pronoun interpretation preferences to consider for future rule-based models.

I.1. Constraints for References

In order to determine references, a number of constraints need to be considered. Many of these are inspired from the excellent lecture slides from Columbia University[1] and Stanford University[2] with Christopher Manning, the author behind several models referred to in this thesis.

[1] `http://www1.cs.columbia.edu/~kathy/NLP/ClassSlides/Slides09/Class19-Pronouns/myref.pdf`

[2] `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1162/handouts/cs224n-lecture10-coreference.pdf`

Number agreement

- John's parents like opera. John hates it (opera)
- John's parents like opera. John hates them (parents)

Person/case agreement

- Nominative: I, we, you, he, she, they
- Accusative: me,us,you,him,her,them
- Genitive: my,our,your,his,her,their

"George and Edward brought bread and cheese. They shared them."

Gender agreement

- Charlie has a Porche. He/it/she is attractive

Syntactic constraints

This is closely related to binding (in linguistics), also called the *binding theory*.

- John bought himself a new Volvo. (himself = john)
- John bought him a new Volvo. (him = NOT john)

Discontinuous sets

Mentions across different sentences.

- John and Anna bought a boat.
- ...
- They were happy with it!

Selectional restrictions with knowledge

- John left his plane in the hangar.

"He had flown it from Memphis this morning". "It" can refer to both the plane and the hangar. Can only fly a plane (from external knowledge), thus the plane is resolved as the correct reference.

I.2. Pronoun Interpretation Preferences

Recency

- John bought a new boat. Bill bought a bigger one. Mary likes to sail it.

Grammatical role

- X went to the dealership with Y. He bought a car
- Y went to the dealership with X. He bought a car
- X and Y went to the dealership. He bought a car

Repeated mention

- John needed a car to go to his new job
- He decided that he wanted something sporty
- Bill went to the dealership with him. He bought a Toyota. (He = John)

Parallel constructions

- John told him (other) he should get over with his (john) obsession with cars
- verb semantics/thematic roles
- John rang Bill. He'd lost the directions he needed
- John shouted at Bill. He'd lost his phone

Pragmatics

- John bought a book for Anna, and so did Bill.

J. Attached Code

In the delivered code, *Tollef-Jorgensen-code.zip*, three folders can be found: *IDUN Cluster*, *Local* and *OPEN_SOURCE_PROJECTS*. A file, *INSTRUCTIONS.md* is located in the root directory, with brief instructions. The *IDUN Cluster* folder contains all files used on the cluster (Själander et al., 2019), in addition to the *slurm* jobs to run each experiment. The *Local* folder holds all files used in development on a personal computer, whereas the final systems are found in the *OPEN_SOURCE_PROJECTS* folder. *README.md* markdown files are within each of the relevant open source projects. Note that almost all systems rely on both licensed datasets and produced data from the Strise Knowledge Graph.

J.1. Datasets

The OntoNotes dataset can not be redistributed, and must be accessed by logging in with an authorized account[3]. The PreCo dataset can be downloaded by filling in a form on

[3] `https://catalog.ldc.upenn.edu/LDC2013T19`

their website[4]. The GUM dataset[5] and LitBank dataset[6] are available on their respective GitHub repositories. To retrieve the Strise data, access must be granted. Data can also be downloaded by using a temporarily valid authorization key which can be provided on request. This key must be added in the following file:

```
Tollef-Jorgensen-code\OPEN_SOURCE_PROJECTS\ElsaVal\Distant
   Supervision\GraphQL Downloader\config.py
```

If the data is downloaded, each of the steps, 1 through 4 must be completed, run in the respective *Jupyter Notebooks* located in

```
Tollef-Jorgensen-code\OPEN_SOURCE_PROJECTS\ElsaVal\Distant
   Supervision\Strise Data Steps
```

J.2. Coreference Evaluations

Experiments with most evaluations computed in this thesis reside in the following file:

```
Tollef-Jorgensen-code\OPEN_SOURCE_PROJECTS\ClEval\
   CorefLiteEvaluation.ipynb
```

This relies on downloading and installing the necessary models SpanBERT[7], NeuralCoref[8] and CoreNLP[9]. Instructions are given in the respective folders.

Before evaluations, the datasets must be converted to the coreflite format. This can be done in

```
Tollef-Jorgensen-code\OPEN_SOURCE_PROJECTS\ClEval\CorefLite\
   Dataset Converters
```

After updating the correct path to the original datasets.

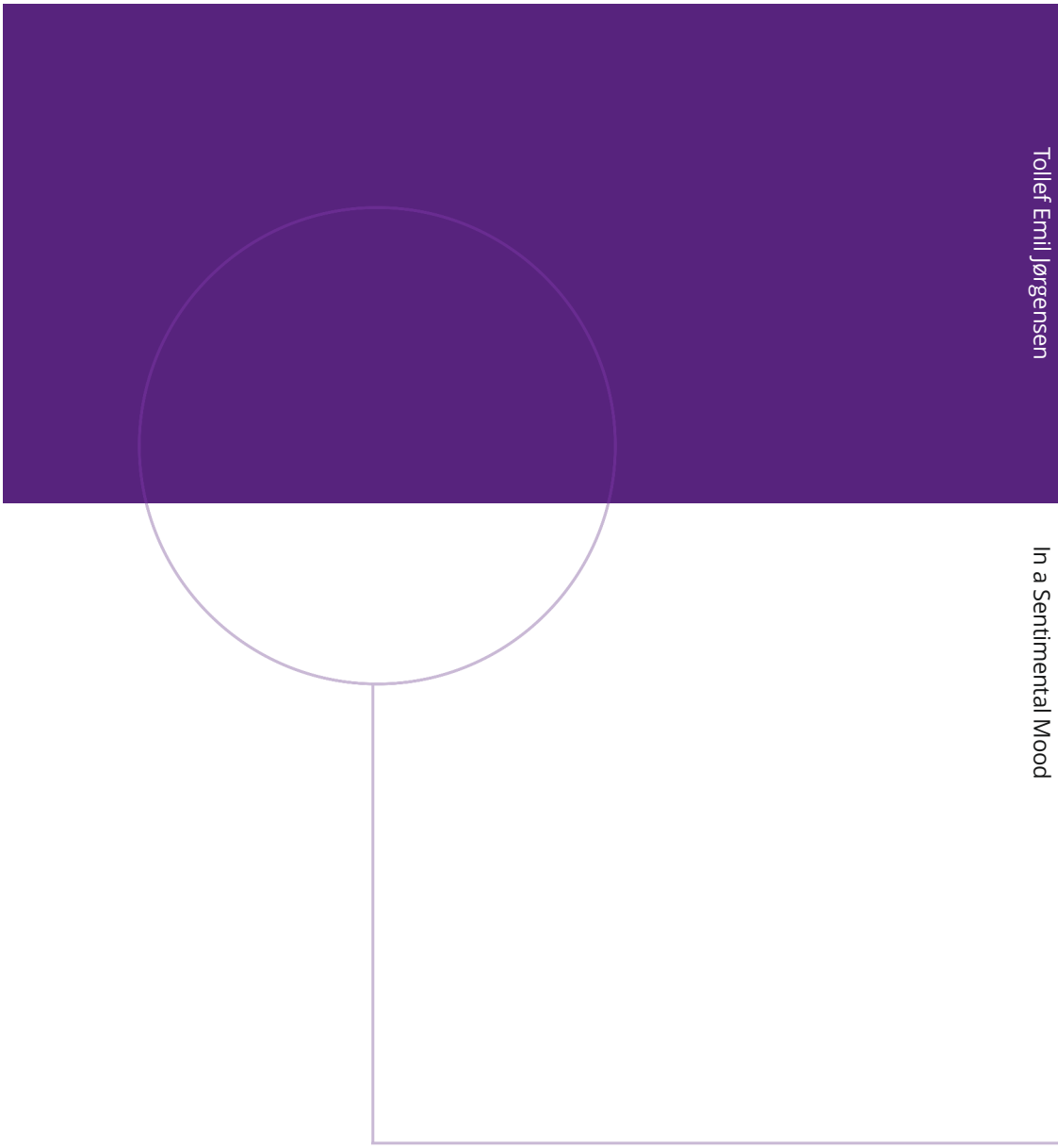
J.3. Entity-level Sentiment Analysis

With properly formatted datasets, placed in the following folder:

```
Tollef-Jorgensen-code\OPEN_SOURCE_PROJECTS\ElsaVal\Models and
   Classification\datasets
```

Experiments can be run by calling the *train.py* file, with arguments found in the *helpers/argparser.py* file. By calling the *train.py* file without arguments, the default setup will be run. Examples can be found in the *.bat* files contained in the *Models and Classification* folder. Any more assistance can be given on request. The open sourced systems are published on GitHub, and links can be found in the Introduction, p. 3. Thanks for reading!

[4] https://preschool-lab.github.io/PreCo/
[5] https://github.com/amir-zeldes/gum
[6] https://github.com/dbamman/litbank
[7] https://github.com/facebookresearch/SpanBERT
[8] https://github.com/huggingface/neuralcoref
[9] https://stanfordnlp.github.io/CoreNLP/



Tollef Emil Jørgensen

In a Sentimental Mood

NTNU
Norwegian University of
Science and Technology