Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Negation Resolution as Dependency Parsing

Elizaveta Sineva
Master thesis

Prüfer:	Prof. Dr. Jonas Kuhn
	Dr. Annemarie Friedrich
Betreuer:	Stefan Grünewald
	Prof. Dr. Jonas Kuhn

Beginn der Arbeit:	14.12.2020
Ende der Arbeit:	14.06.2021

Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet. Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen. Sie ist weder vollständig noch in Teilen bereits veröffentlicht. Die beigefügte elektronische Version stimmt mit dem Druckexemplar überein.

(I declare that I have developed and written the enclosed thesis completely by myself and that I have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere. The enclosed electronic version is identical to the printed versions.)

(Elizaveta Sineva)

Abstract

Negation resolution refers to the task of detecting negation in text, as well as the specific spans of text which are negated. Past work has mostly modeled it as a sequence tagging task; however, recently the idea of modeling it as dependency parsing has been explored. In this thesis, we look at different ways to encode negation data as dependency graphs. We make use of a previously suggested encoding by Kurtz et al. (2020) and propose four novel encoding schemes. We apply a transformer-based dependency parser to the data and analyze the effects of its performance on the task. We test the five data encodings and find that the nested encoding performs best in most cases. We use a sequence-tagging baseline in contrast to our approach and show that our system with the use of an embedding model fine-tuned for syntactic dependency parsing outperforms the baseline on in-domain experiments. In order to provide fair comparison with the baseline, we create an evalution script covering a wide range of existing metrics used for negation detection by different works.

Contents

List of Figures

List of Tables

1 Introduction

Negation resolution is a task focusing on detecting negation instances in text. It is applicable in various natural language processing tasks, for example, information extraction, especially in the biomedical domain (Szarvas et al., 2008; Mehrabi et al., 2015) where the factuality and reliability of information is essential. It can be useful for sentiment analysis (Wiegand et al., 2010; Moore and Barnes, 2021) as negation cues can alter the meaning of a given statement and hence indicate a different emotion from that of a non-negated statement. Another area of application is machine translation (Fancellu and Webber, 2015; Bentivogli et al., 2016), where the correct identification of the negated part of the sentence is important for its correct translation.

Negation detection includes recognizing a negation signal, or a *cue*, and identifying the part of the text that is affected by that cue (its *scope*). The task may also include negated *event* resolution. The approaches to solving the task in question range from rule-based to different machine- and deep-learning-based ones. While many systems achieve good results in cue detection, scope resolution has proven to be a more difficult task. Another challenge of the task is divergence in the annotation guidelines of different datasets, which makes it harder to generalize over domains. Furthermore, the existence of a great number of evaluation metrics without universally accepted ones makes the task even more complicated. With different works using different metrics to evaluate their systems, it is also a challenge to compare different systems.

The boundaries of the scope affected by the cue are syntactically grounded, which led to many negation resolution systems incorporating syntactic dependency information. Many rule-based approaches to negation resolution rely on dependency parsing (Sanchez Graillet and Poesio, 2007; Sohn et al., 2012; Mehrabi et al., 2015). Moreover, machine learning systems for negation detection often employ syntactic information as additional features (Read et al.,

2012; Lapponi et al., 2012; Cruz Diaz et al., 2015). However, there are few research works focusing on applying a dependency-parsing based approach directly. Thus, our interest lies in finding a solution by means of dependency parsing algorithms. In their recent work on negation resolution, Kurtz et al. (2020) apply a parsing model to negation data which achieves state-of-the-art results. Following their idea, in this thesis we further explore the capabilities of dependency parsing methods and examine their impact on the negation detection task by framing them as dependency parsing. The essence of the approach is reformulating the task as a set of binary relations within a sentence, or, in other words, shaping the given data into dependency trees. We experiment with different ways of encoding the data in such a way by replicating the representation created by Kurtz et al. (2020) and developing four mappings ourselves. The categorization of the relations is adapted to the task by using *cue*, *scope* and *event* as the labels, as opposed to using the grammatical categories of the original task. We apply the parser to different dependency-like representations of the same data and study the performance systematically using a graph-based dependency parser employing pre-trained transformer-based language models (Grünewald et al., 2021) as our main tool. We compare the results for different representations, demonstrating that the nested tree representation performs best in most cases, with the direct cue-to-scope representation coming in as the second best. We implement our own evaluation script in order to cover a range of metrics used in different works, thereby making a fair comparison of different systems possible. We compare a sequence-tagging based baseline to our models and show that the use of a dependency parsing in combination with an embedding model fine-tuned on syntax outperforms the baseline.

The thesis is structured as follows. Section 2 reports the prior research relevant for the topic. Section 3 describes the datasets used in this work. Section 4 provides the details on the applied parser and the data representations. Section 5 describes the performed experiments, the settings of the experiments and their results. Section 6 provides an analysis of the results

with respect to the settings that they were obtained in, as well as an outlook. Section 7 concludes the thesis.

2 Related Work

In this section, we provide background information with regard to the negation resolution, as well as provide an overview of the work related to the negation resolution, different approaches to the task, and the systems that we used as our baseline.

2.1 Background

Negation is a linguistic device that is used to reverse the meaning of a sentence or a part of a sentence, or to convey the falseness of information. A negation instance is comprised of a *cue*, which is a "word that expresses negation" (e.g. *no*, *not*, *without* etc.), and its *scope*, which is "the part of a sentence that is affected by the negation cues" (Morante and Blanco, 2012; p. 268). Some negation datasets also include an *event* that is negated by the cue. Morante and Blanco (2012) define *event* as "the main event or property actually negated by the negation cue" (see an example in Figure 1).

...	I	noted	that	there	were	**no**	other	footsteps	...
				Scope	*Scope*	***Cue***	*Scope*	*Event*	

Figure 1: Example of a negation instance (taken from *ConanDoyle-neg*).

A cue can be expressed in a variety of ways. It can appear in a form of a negation word (e.g. *never*) or a series of words (e.g. *by no means*), or it can be a negation affix (e.g. *in*frequent). Negation can be expressed in far less obvious ways, e.g. with the help of some polysemic verbs such as *fail* or *avoid*. Such cases pose an additional challenge as these are not precise negation signals and may refer to an action rather than negation (cf. *I failed to leave the room.* vs. *I failed the exam.*) Another rexample of a false negation cue would be such phrases as *no doubt* (e.g. *He is no doubt a genius.*), *not only*, *no wonder* etc. (Carrillo de Albornoz et al., 2012; p. 284)

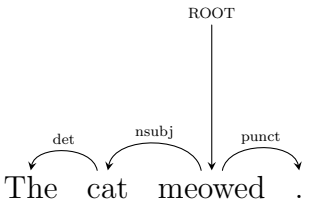


Figure 2: Example of a syntactically parsed sentence.
det = determiner, nsubj = nominal subject, punct = punctuation.

Dependency parsing aims at structuring a sentence by representing the (syntactic) relations of its words in the form of a tree or a graph. Defining dependency, Nivre (2005) (p.3) states that "the syntactic structure of a sentence consists of binary asymmetrical relations between the words of the sentence", i.e. hierarchical relationships between pairs of words or tokens which are assigned the roles of a head and a dependent. The relations are also categorized in accordance with their grammatical function, including such categories as subject, direct or indirect object and others (Jurafsky and Martin, 2020; p. 281-282). For example, in Figure 2 the head verb *meowed* governs the dependent noun *cat*, with the role of the dependent in relation to its head being nominal subject (`nsubj`). One of the approaches to dependency parsing is *graph-based parsing*, the main idea of which is to "search through the space of possible trees for a given sentence for a tree (or trees) that maximize some score" using techniques from the graph theory (Jurafsky and Martin, 2020; p. 296).

2.2 Previous Approaches

Early work on negation resolution mainly focused on various rule-based systems. One of the first negation detection systems, NegEx (Chapman et al., 2001), relies solely on domain-specific regular expressions and lexical cues to solve the task. However, given that the negation scope boundaries are related to the syntactic structure of the sentence it is found in, a number of works use syntactic information when defining the rules for their systems.

Sanchez Graillet and Poesio (2007) extract scopes via a pattern recognition system using lexical information and syntactic dependency parses. Sohn et al. (2012) develop a set of dependency-parsing based rules and single out syntactic patterns typical for negated constractions. They note that the use of the syntactic approach helps to capture the scope when it is located far away from the cue in the sentence, as it is not restricted by the word distance in contrast to a raw regular-expression approach (Sohn et al., 2012; p. 7). Carrillo de Albornoz et al. (2012), participants of the *SEM 2012 Shared Task on cue detection and scope resolution (Morante and Blanco, 2012), make use of syntax trees, or constituency trees, to determine the scope boundaries. Rule-based systems are used in later years as well. Mehrabi et al. (2015) build upon the output of NegEx by further filtering out false predictions using syntactic dependency information obtained from Standford Dependency Parser. NegBio (Peng et al., 2017) uses dependencies as well to establish patterns typical for negation.

Rokach et al. (2008) transform the rule-based regular expression approach into a machine learning problem, using a tree classifier to automatically learn regular expression patterns. Morante and Daelemans (2009) combine rule-based and machine-learning based methods for the cue detection. First, they look for unambiguous cues using a list of negation signals extracted from the training corpus, such us *no*, *not*, *abscence* etc. (Morante and Daelemans, 2009; p. 24) Next, they apply a memory-based classifier to label the tokens that were not selected in the pre-processing step. For the scope resolution, they use a combination of three classifiers: a memory-based classifier, a Support-Vector Machine (SVM) and Conditional Random Fields (CRFs). Councill et al. (2010) utilize a CRF classifier by feeding various syntactic dependency information as features for the scope detection. Cruz Diaz et al. (2015) apply an SVM classifier based on a Radial Basis Function (RBF) kernel for the negation resolution using Beginning-Inside-Outside (BIO) encoding. They avoid incorrect predictions as the majority class with the help of Cost-Sensitive Learning (CSL). They note that syntactic features seem

redundant for the cue detection (Cruz Diaz et al., 2015; p. 17); however, they use the syntactic information in the scope detection phase and show improvement over systems without the incorporation of syntax.

Li et al. (2010) introduce the idea of framing negation resolution as a shallow semantic parsing problem. They map the negation information onto semantic constituency trees and use a pruning algorithm to remove the constituents that are least likely to refer to negation. They then apply an SVM classifier to predict the negation scope, and process the results further to deal with discontinuous predictions.

Many of the participants of the *SEM 2012 Shared Task on negation detection turn to machine-learning approaches, incorporating syntactic features as well. Read et al. (2012) apply an SVM classifier for cue detection and scope resolution. Fot the latter, they make use of the syntactic constituents provided in the dataset used in the Shared Task and apply an SVM-based ranking of the consitituents. The system achieves the best global score and event score in the task. Lapponi et al. (2012) use an SVM classifier for cue detection and a CRF sequence tagger with syntactic dependency information as features for scope and event resolution. They use Stanford Dependency Parser and Maltparser for the two versions of the classifier. The system performs best for the scope resolution (the best scopes with no cue match metric; see Section 5.1 for a detailed description of the metrics). While White (2012) use a rule-based system for cue detection, they apply a CRF sequence labeling system for scope and event resolution. They use the provided syntax tree information, among other features.

The current state of the art systems are mainly based on neural networks. Fancellu et al. (2016) compare a Feedforward Neural Network (FNN) and a Bidirectional Long Short-Term Memory (BiLSTM) for scope resolution. They experiment with different settings, namely adding cue information, pretrained embeddings and Part-of-Speech (POS) tag information. They show that a BiLSTM model with a combination of the three aforementioned features yields the highest scores. They note that the CRF-based system used

by White (2012) performs better at exact scope matching, as they make use of the syntax trees which are helpful for detecting scope boundaries (Fancellu et al., 2016; p. 502). Fancellu et al. (2018) build on their BiLSTM model and create a cross-lingual negation resolution system by using universal dependencies as their input to the model. McKenna and Steedman (2020) adjust the Global Belief Tree Recursive Neural Network (GBTRNN) by Paulus et al. (2014) to the task of negation detection. They use the syntax trees as the input to the model. The model only uses the information about the cue and the syntactic constituents, while the lexical information stays hidden.

2.3 Baseline Approaches

A recent work by Khandelwal and Sawant (2020) uses a transformer-based classifier, NegBERT, to train two models: a cue detection model and a scope resolution model. They employ different labels for different cue types (affixational, one-word, multi-word and not a cue). For scope resolution, they use binary labels (in / out of scope). They use BERT (Devlin et al., 2019) as their embedding model. They also utilize BERT to tokenize the words into subwords and train the model on the tokenized data. For scope resolution, they only provide the sentences that contain the cues to the model, encoding the cues either by replacing them with special tokens (the "replace" method), or adding a special token in front of the cue (the "augment" method). In both cases the special token reflects the type of the cue it is related to. The prediction happens in two phases: first predicting the cues and then the scope. They postprocess the predictions to combine the subword-level tokens into word-level tokens. Britto and Khandelwal (2020) build on NegBERT and expand the task to uncertainty detection, phenomena often annotated in corpora in parallel wih negation. They compare the influence of different embedding models on the tasks, specifically BERT, RoBERTa (Liu et al., 2019) and XL-Net (Yang et al., 2020). They also elaborate on their post-processing step, comparing the two options of choosing a label for a word-level token out of

subword-level labels: "average" (average the scores over all labels and output the label with the greatest score) and "first token" (output the label of the first subword-level token). Khandelwal and Britto (2020) approach the two tasks by using the same architecture, but in multitask settings, training for both negation and speculation at once. We use this system as our baseline to contrast our dependency-based neural model with a sequence-tagging based model.

Kurtz et al. (2020) suggest an approach that uses dependency graph representations for negation resolution. They convert negation cues into the dependency heads of their corresponding scope (dependents), with a small set of relation labels referring to the scope (`S`) itself, the event (`E`) within the scope and an additional label for the multi-word (`M`) cues (e.g. *neither... nor...*). The cue (`CUE`) itself is governed as well by the root node (see Figure 7 of Section 4.2). The architecture for the suggested approach uses several types of embeddings fed into a BiLSTM. FNN is applied to the output to obtain the representation vectors corresponding to possible heads and dependencies, which are then scored using a bilinear model (Dozat and Manning, 2018) to determine the most probable cues and scope spans. We build on the work of Kurtz et al. (2020) and approach the negation detection as a dependency parsing task, using their negation data representation and developing four of our own novel representations.

3 Data

A number of datasets from different domains have been annotated with negation information. Earlier work is bound in particular to the biomedical domain, with the detection systems often paying attention to specific medical terms being negated. *BioScope* (Szarvas et al., 2008) is a frequently used biomedical dataset annotated for negation. The concentration on the medical domain is due to it being one of the main areas of application of negation detection systems, as well as simply lack of data in other domains. The domain coverage of negation corpora has since expanded, with Morante and Daelemans (2012) introducing *ConanDoyle-neg*, a corpus of fictional literature annotated for negation, as well as Konstantinova et al. (2012) annotating with negation a corpus of reviews in various domains (*SFU Review*). However, annotation guidelines differ significantly between corpora, meaning that models trained on one domain tend to generalize poorly for use on other domains. In this section we describe the three aforementioned datasets that we use for negation detection, their annotation schemes and how they differ (see Table 1 for an overview).

3.1 ConanDoyle-neg

The ***ConanDoyle-neg* (CD-SCO)** dataset (Morante and Daelemans, 2012) comprises several literary texts written by Conan Doyle annotated with respect to negation and its scope. The dataset was created for negation cue and scope detection (Task 1) in the *SEM 2012 Shared Task (Morante and Blanco, 2012), with a total size of 5,520 sentences. It is composed of the novel *The Hound of the Baskervilles* used as a training set (66% of the data), *The Adventure of Wisteria Lodge* as a development set (14.3%) and both *The Adventure of the Red Circle* and *The Adventure of the Cardboard Box* as test sets (19.7%). *ConanDoyle-neg* annotations include the cues, the negated event corresponding to the given cue and the scope of negation (which ex-

Dataset	***ConanDoyle-neg***	***BioScope*** *Abstracts*	*Full papers*	***SFU Review***
Source	Morante and Daelemans (2012)	Szarvas et al. (2008)		Konstantinova et al. (2012)
Domain	fiction writing	biomedical		review
Sentence #	5,520	11,871	2,670	17,263
Negation sentence #	1,227	1,597	339	3,117
Negation instance #	1,421	1,719	376	3,518
Cue is a part of the scope	no	yes		no
Includes discontinuous scopes	yes	no		yes
Includes events	yes	no		no
Annotates negation affixes	yes	rarely, with the whole word as a cue		yes, but with the whole word as a cue
Tokenized	yes	no		yes
File format	CoNLL	XML		XML

Table 1: Dataset overview.

cludes the cue, but includes the event as as its part). The annotation scheme allows discontinuous scopes:

*...(those) not **infrequent** (occasions when he was up all night)...*

The negation cues in the dataset can be divided into three types:

- One-word cues (e.g. *no*, *not*, *without*)
- Multi-word cues (e.g. *neither... nor*, *by no means*, *on the contrary*)
- Negation affixes
  - Prefixes (5): un-, in-, im-, ir-, dis- (e.g. *in*advertently, *dis*pleasure)
  - Suffix (1): -less (e.g. help*less*[ly])

The dataset is in CoNLL format (further **SEM 2012 format*) consisting of 8+ columns. Columns 1-7 contain the file name, the sentence and token numbers, the word form, the lemma, the part-of-speech tag and a constituency tree in bracketed form. The rest of the columns contain the information about negation in the sentence according to the following principles:

b01	153	0	I	I	PRP	(S(NP*)	_	_	_	_	_	_	_
b01	153	1	trust	trust	VBP	(VP*	_	_	_	_	_	_	_
b01	153	2	,	,	,	*	_	_	_	_	_	_	_
b01	153	3	sir	sir	NNP	(NP*)	_	_	_	_	_	_	_
b01	153	4	,	,	,	*	_	_	_	_	_	_	_
b01	153	5	that	that	IN	(SBAR*	_	_	_	_	_	_	_
b01	153	6	I	I	PRP	(S(NP*)	_	I	_	_	_	_	_
b01	153	7	have	have	VBP	(VP*	_	have	_	_	_	_	_
b01	153	8	not	not	RB	*	not	_	_	_	_	_	_
b01	153	9	inadvertently	inadvertently	RB	(ADVP*)))))	_	inadvertently	_	in	advertently	_	
b01	153	10	–	–	:	*	_	_	_	_	_	_	_
b01	153	11	”	”	”	*)	_	_	_	_	_	_	_

Figure 3: Example sentence from *ConanDoyle-neg* with 2 negation instances.

(a) When there is no negation in the sentence, the eighth column contains ***;

(b) For every negation instance in the sentence three new columns are added to the seven existing columns; for example, if there are two negation instances, the sentence will have thirteen columns in total, with columns 8-10 corresponding to the first negation instance, and columns 11-13 to the second one. The first negation column of an instance corresponds to the cue, the second to the scope and the third to the event (Figure 3).

3.2 BioScope

BioScope (Szarvas et al., 2008) is a dataset concentrating on negation and speculation in biomedical texts. It consists of three subcorpora: biological paper abstracts (11,871 sentences) from the GENIA Event corpus (Kim et al., 2008), full biological papers (2,670 sentences) and radiology reports (we did not include the latter in this work). The dataset consists of negation and speculation cues and their corresponding scopes. Unlike the aforementioned *ConanDoyle-neg* dataset, *BioScope* is not annotated for events. Furthermore,

```
<sentence id="S1.15">
     Our result
          <xcope id="X1.15.3">
               <cue type="speculation" ref="X1.15.3">suggests</cue>
               that
               <xcope id="X1.15.2">the unknown amino acid encoded by stop codons does
                    <xcope id="X1.15.1">
                         <cue type="negation" ref="X1.15.1">not</cue>
                         exist
                    </xcope>,
                    <cue type="speculation" ref="X1.15.2">or</cue>
                    its phylogenetic distribution is rather limited
               </xcope>
          </xcope>
     , which is in agreement with the previous study on tRNA.
</sentence>
```

Figure 4: Example sentence from *BioScope*.

it only includes one-word and multi-word cues, leaving the negation affixes unmarked.

The following XML format is used for the corpus annotation. Each sentence is encompassed within a `sentence` node, with chunks of text marked as `cue` or `xcope` (scope) when necessary. The scopes are always continuous, with cues being a part of their scope. Negation and speculation cues are distinguished through the inclusion of the attribute `type`. Each scope element has an `id`, and each cue element has a `ref` ID referring to the corresponding scope (see Figure 4). The corpus is not tokenized.

3.3 SFU Review

The ***Simon Fraser University Review*** corpus (*SFU Review*, Taboada et al., 2006) is a corpus of reviews in eight different domains (books, cars, computers, cookware, hotels, movies, music, phones). Each domain includes 50 reviews (25 negative and 25 positive). The total size of the corpus is 17,263 sentences. The corpus was annotated for negation and speculation by Konstantinova et al. (2012). Like *BioScope*, it is annotated for cues (one-word and multi-word) and their scopes and does not mark events. The words with negation affixes are marked as negation cues in *SFU Review*, but without separating the affix itself from the word.

```
<SENTENCE>
    <W>I</W>
    <cue ID="0" type="negation">
        <W>do</W>
        <W>n't</W>
    </cue>
    <xcope ID="21">
        <ref COMMENTS="" ID="23" SRC="0"></ref>
        <W>know</W>
        <W>why</W>
        <W>you</W>
        <cue ID="1" type="speculation">
            <W>would</W>
            <W>n't</W>
        </cue>
        <xcope ID="20">
            <ref COMMENTS="" ID="22" SRC="1"></ref>
            <W>read</W>
            <W>this</W>
        </xcope>
    </xcope>
    <W>...</W>
</SENTENCE>
```

Figure 5: Example sentence from *SFU Review*.
I don't know why you wouldn't read this...

The corpus uses an XML format similar to *BioScope*, and bases its annotation on the *BioScope* guidelines. However, there are certain deviations which are important to take note of. The cues are not considered to be part of the scope, with the scopes now referring to the cue IDs via an additional `ref` tag and its attribute `SRC` (Figure 5). This approach also results in allowing discontinuous scopes in the data. As multi-word cues have a separate ID for each of its parts, the scope reference ID includes all of the cue IDs separated by a space (e.g. for a multi-word cue with IDs "39" and "40", the scope `SRC` ID will be "39 40"). Unlike *BioScope*, *SFU Review* provides tokenization information by encompassing every token within `W` tags.

4 Methodology

In this section, we describe our approach to the task, as well as details on the architecture used. We provide a description of different ways to encode negation resolution as dependency trees, including some of their advantages and drawbacks.

4.1 Dependency Parser

We use the graph-based dependency parser **STEPS** (Stuttgart Transformer-based Extensible Parsing System) by Grünewald et al. (2021). The parser (see Figure 6) uses the biaffine architecture suggested by Dozat and Manning (2017), a standard approach to graph-based dependency parsing. Every token is transformed into two representations as a potential head and a potential dependent applying feed-forward neural networks to the token embeddings. The biaffine classifier then provides scores for all possible head-dependent relations with the help of the obtained representations. We apply the unfactorized approach (Dozat and Manning, 2018) to the processing of the scores, which makes use of a single classifier scoring all head-dependent-label combinations, with the absence of an arc between a head and a dependent encoded as a `null` label. The system outputs the highest-scoring label for every head-dependent pair, with additional post-processing for each of the different data representations. The post-processors implemented for specific representations will be described in Section 4.2.

The inputs to the system are composed with the help of pre-trained contextual word embeddings. We utilize multilingual XLM-R(oBERTa) (Conneau et al., 2019) based on the ideas of RoBERTa (Liu et al., 2019), as well as a version of XLM-R fine-tuned for syntactic dependency parsing (Grünewald et al., 2021) as our language models. The final token embeddings are computed through the application of a layer attention mechanism following the methods of Kondratyuk and Straka (2019).

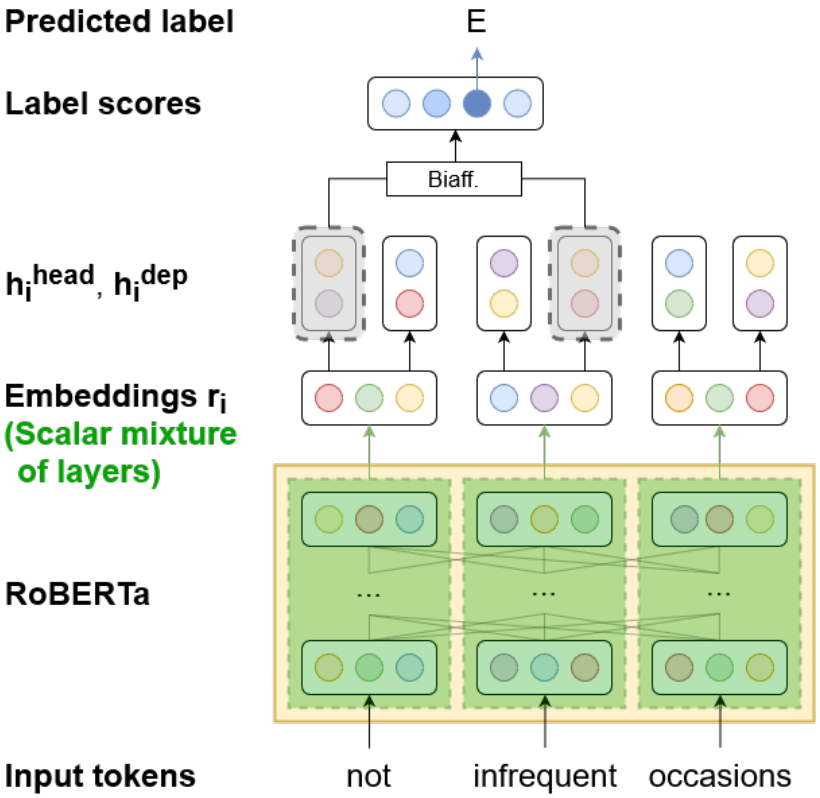


Figure 6: STEPS architecture (Grünewald et al., 2021).

The system also supports multitask training based on the approach of Kondratyuk and Straka (2019). The coefficients for the scalar mixture of the transformer layers, as well as the biaffine scorer weights are learned separately for each task.

STEPS is built in Python with PyTorch (Paszke et al., 2019) as its base. Huggingface Transformers library (Wolf et al., 2019) is used to work with the transformer-based models.

4.2 Dependency Parsing Format

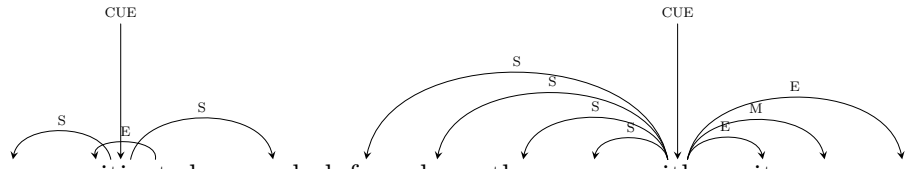
In order to approach negation resolution as a parsing problem, we change the representation of the data. Following the ideas of Kurtz et al. (2020), we

transform the data into dependency graphs, where the cues are represented as the root elements of the sentence, and the scope elements are governed by the cue in different ways. The root relations to the cues are labeled as `CUE`. In the case of multi-word cues, the first token of the cue is the root, with the rest of the tokens governed under the relation label `M` (for Multi-word). The relations corresponding to event within the scope (if applicable to the data) and the scope itself are labeled as `E` and `S` respectively. Using this labeling scheme, we apply different mappings of the given data to the new format in order to explore their influence on the training process.

We use a CoNLL format to store the data. The format requires tokenization, which we keep the same as in the original datasets with a few exceptions. As *BioScope* is not tokenized, we utilize NLTK for tokenization with some additional post-processing rules for punctuation and URLs. For *SFU Review*, we also break into tokens the original tokens that contained two words and a newline break. To keep most of the data introduced in the original datasets (mainly *ConanDoyle-neg*), we include five columns in the new format: token number, word form, lemma, part-of-speech tag and dependency information in the form *head : relation label* (Figure 7b). The pipe symbol (|) is used to separate all of the relations for tokens with more than one head.

4.2.1 Direct Cue-to-Scope Mapping

In this mapping (Kurtz et al. (2020) mapping), the cue acts as the head of all other elements in the scope. For multi-word cues, the first cue token (`CUE`) governs the rest (`M`). A special case of a cue-event dependence occurs in the words with negated affixes (e.g. *un*mitigated), where the affix (*un-*) acts as the cue and the rest of the word (*mitigated*) as the event. Such cases are specific to the *ConanDoyle-neg* dataset and are handled with the help of a self-looping arc labeled `E` within the cue word (Figure 7). Post-processing is used for predictions in this format that remove all scope and event arcs that are not headed by a cue. The predictions are then converted into a *SEM



(a) Visualized graph.

...				
9	,	,	,	_
10	an	an	DT	11:S
11	unmitigated	unmitigated	JJ	0:CUE|11:E
12	scoundrel	scoundrel	NN	11:S
13	for	for	IN	17:S
14	whom	whom	WP	17:S
15	there	there	EX	17:S
16	was	be	VBD	17:S
17	neither	neither	DT	0:CUE
18	pity	pity	NN	17:E
19	nor	nor	CC	17:M
20	excuse	excuse	NN	17:E
21	.	.	.	_

(b) CoNLL representation.

Figure 7: Direct cue-to-scope mapping (example taken from Kurtz et al. (2020), *ConanDoyle-neg* dataset).

2012 format by interpreting every token in connection to the label of the arc it is headed by. The mapping is easily interpretable as it makes direct connection between the cues and scopes. However, a possible flaw is a lack of a more specific pattern for a parser to learn.

4.2.2 Nested Mapping

Nested mapping is based on the direct cue-to-scope mapping but adapted to nested scopes. A nested scope refers to the scope of a negation instance

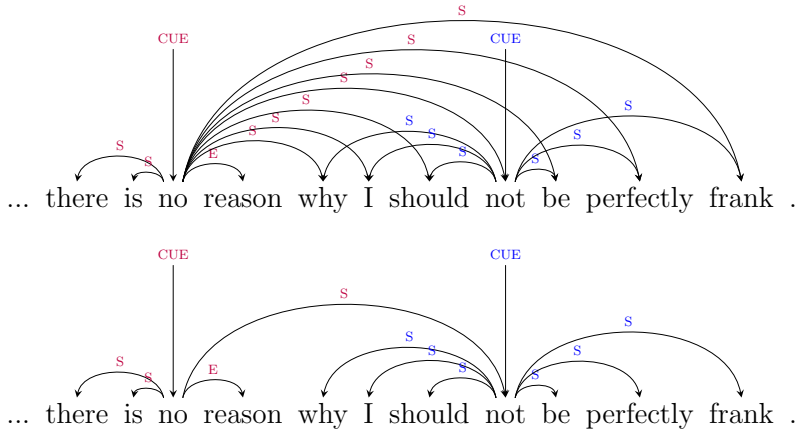
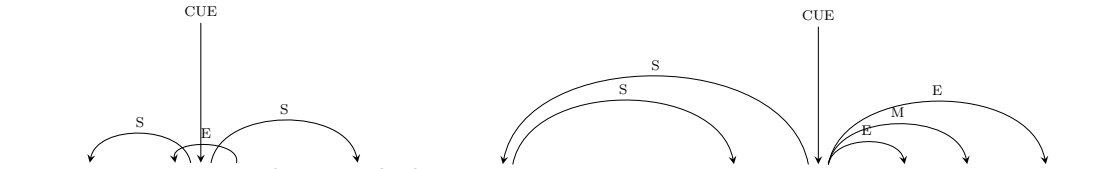


Figure 8: Direct (upper) vs. nested (lower) encoding (example taken from *ConanDoyle-neg*).

that is a part of the scope of another instance. For an example, see Figure 8: the scope of the cue *not* is nested in the scope of the cue *no*. All three datasets allow nested scopes. If a cue is part of the scope of another cue, the scope of the first cue will be nested in the scope of the second cue. Thus, we can modify the direct cue-to-scope mapping to omit redundant arcs by only keeping the arcs that do not belong to a nested scope within the modified scope. For the prediction, the same post-processing step as for direct mapping is performed, with the scope not headed by the cue removed. The process of conversion of the output is also similar, with an additional step of adding all of the child element's scopes and events to a head cue as well. Reducing the arcs allows us to reduce the information necessary to learn. It also reduces the number of labels associated with a token, when the token is an event for one of the cues; here, it is only marked as an event and is automatically recognized as a scope for the parental cues.



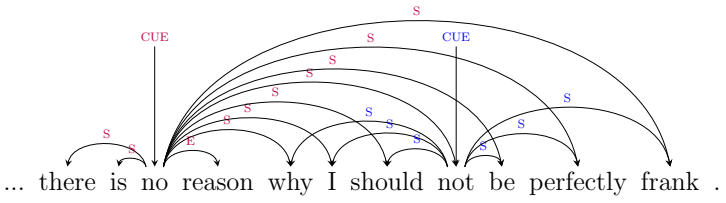
(a) Spans without an end arc. (b) A span with beginning and end arcs.

Figure 9: Span-like mapping (example taken from *ConanDoyle-neg*).

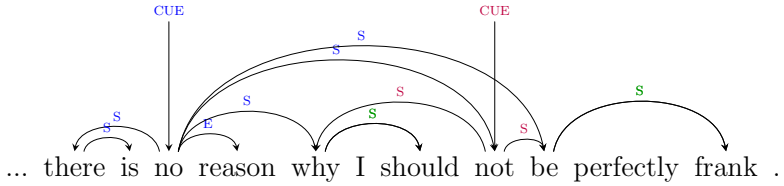
4.2.3 Span-like Mapping

Span-like mapping differs from the previous ones mainly in the encoding of the scopes. This mapping was inspired by Yu et al. (2020), who use dependency parsing for named entity recognition. The idea is to mark the beginning and the end of the scope span rather than marking its every element, capturing a "window" of the scope and relying more on word distance from the cue. However, direct encoding of the spans as an arc from the cue as the beginning and the arc from the beginning to the end has shown to be problematic. To avoid ambiguity when decoding and to stay consistent, the following set of rules was developed:

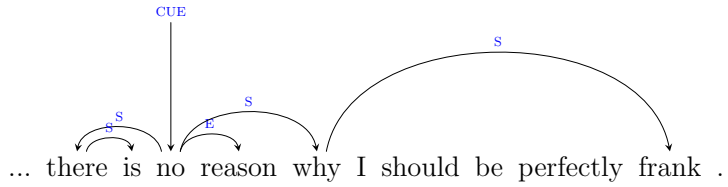
- The cue acts as the head of the token that is the beginning of the scope span. In turn, the beginning acts as the head of the token that marks the end of the span (e.g. Figure 9b);
- If the span is only one token long, the arc to mark the end of the span is not created (e.g. Figure 9a);
- The cues (including the multi-word parts) are excluded from their scopes, since a cue is not a part of its scope, as well as because a scope span cannot start from its cue as it would cause a confusion with an affixational self-loop;
- A scope span cannot start from an event as well as in that case it might be confused with an arc that is only marking the event and not



(a) Direct mapping.



(b) Sharing arcs in span-like mapping (green).



(c) No sharing arcs in span-like mapping (only one cue).

Figure 10: Arc sharing in span-like mapping (example taken from *ConanDoyle-neg*).

the start of the scope. For that reason, if an event is the first element of a scope, the next scope element is marked as the beginning of the span (e.g. Figure 10c). Events not surrounded by other scope elements are excluded from the scope as well as they provide information about being a part of the scope on their own (e.g. Figure 9b). Otherwise, events are included in the scope;

- The cue can govern several scope spans:
  - In the case of discontinuous scopes, every part of the scope is encoded as a separate span;
  - In order to avoid ambiguous span boundaries, a token marking the beginning of a scope span always has only one dependent marking its end. Thus, when two scopes have the same beginning,

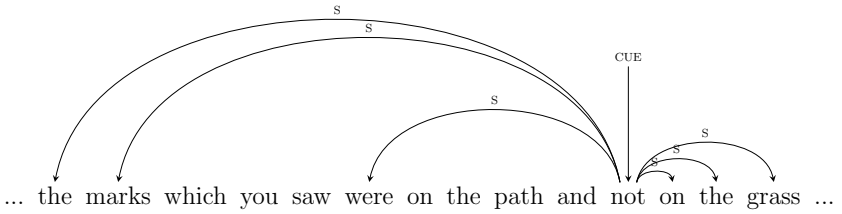
the longer scope is divided into two spans, sharing one span with the shorter scope with their overlap and creating an additional span for the non-overlapping part (see Figure 10);
  - The scope is also divided into several parts if it starts with the cue of a nested scope. In that case, the cue token is encoded as a part of the scope with a separate arc, and the part that comes after is encoded separately, e.g. Figure 10b (in case the token is also an event, only the event label is kept, e.g. Figure 9a).

Unfortunately, the necessity of additional rules makes the mapping rather complicated. This could potentially be resolved with an alternative mapping, with both the start arc and the end arc coming from the cue and having special labels to distinguish between those. This mapping is not used in this thesis due to the time constraints.

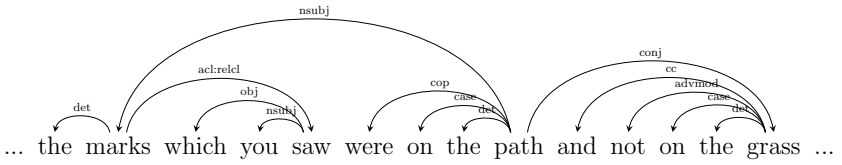
4.2.4 Syntactic Mapping

Syntactic mapping builds on a syntactic dependency parse of the sentence. The objective is to create syntax-based patterns in the representation while encoding negation information. This is achieved via a set of rules:

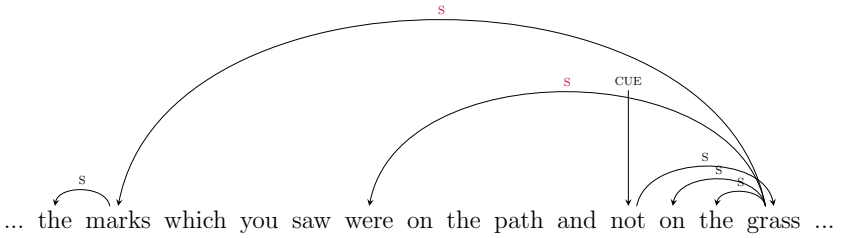
- For every cue in the sentence the incoming syntactic arcs are reversed, i.e. their heads are set as their dependants as long as the head is a part of the scope of the given cue ("reverse rule"; e.g. an arc between *not* and *grass* in Figure 11c);
- For every token in the scope that is not affected by the reverse rule, the syntactic arc is kept as long as the head of the given scope token is a part of the same scope;
- In the case of affixational negation, the part of the word that belongs to the scope is encoded as a self-loop as in the other mappings;
- Based on the principles used to assign the head to the scope tokens that do not fall into one of the above cases, the mapping is divided into two separate representations:



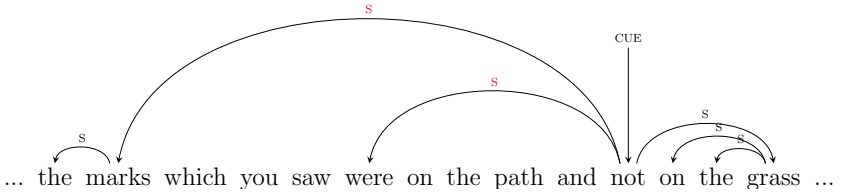
(a) Direct mapping.



(b) Predicted dependencies.

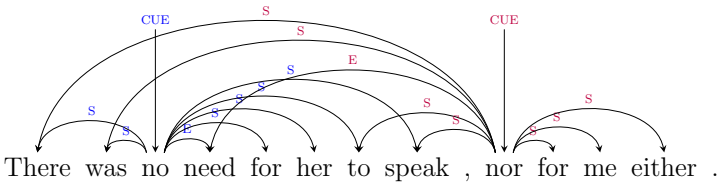


(c) Syntactic mapping.

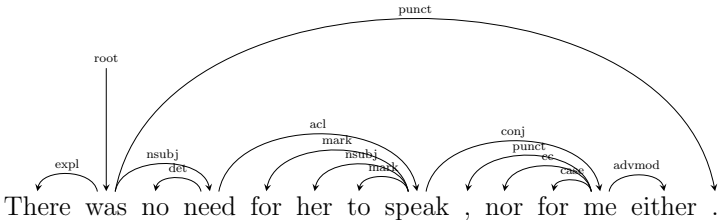


(d) Syntactic-direct mapping.

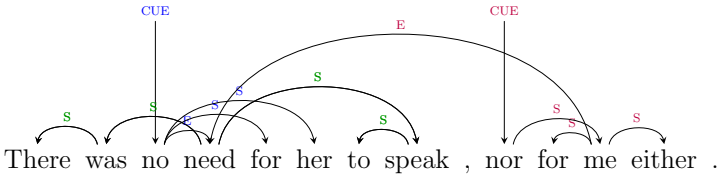
Figure 11: Syntactic vs. syntactic-direct mapping (example taken from *ConanDoyle-neg*).



(a) Direct cue-to-scope mapping.



(b) Predicted dependencies.



(c) Syntactic mapping with shared arcs (green).

Figure 12: Shared scope in mappings (example taken from *ConanDoyle-neg*).

Syntactic:

- The tree path to the given scope token is explored until another parental element that belongs to the same scope is found and appointed as the head of the given token in the scope;
- If the root is reached before a head is assigned, the last token affected by the reverse rule is assigned as the head (see Figure 11c);
- To avoid ambiguous arcs, the cue of the given scope is assigned as the head if the last token is a cue of another scope.

Syntactic-direct:

- The tokens in the scope are directly linked to their cue as in the direct cue-to-scope mapping (see Figure 11d).

- In the case of nested scopes, all tokens of the nested scope are linked to the outer cue via the cue of the nested scope as in the nested mapping;
- In the case of several partially overlapping scopes, a token that belongs to the scope of more than one cue is only allowed to be the head of a token that also belongs to the same scopes (Figure 12);
- In order to avoid cycles, the non-shared arc is connected to the highest shared head in case of shared scopes (e.g. from *me* to *need* rather than to *speak* in Figure 12c).

The predictions with this mapping are post-processed to exclude loops from the dependency graph, as well as to exclude any scope predictions that are not linked to a cue. This mapping's advantage is that it provides a pattern natural to the language. Unfortunately, it was not possible to keep every arc exactly the same as in the actual dependency parses. As the datasets used in this work do not include syntactic information, we used the syntactic dependencies obtained from a model of the STEPS parser (see section 4.1). This could lead to errors as well due to the possibility of incorrect parses of the sentences.

5 Experiments

In this section, we will explain the metrics that we use for evaluating our results, as well as our experimental setup.

5.1 Evaluation Metrics

Previous work has made use of a wide range of metrics for the evaluation of negation detection. A negation instance can be evaluated on the correctness of the prediction of its cue, scope and event, as well as all of its elements together ("full" score). Furthermore, the elements can be evaluated on the token level or the scope level. For the *token* level, the predicted label for every token of a scope (or cue/event) is considered individually. The metric indicates:

- How many of the tokens were correctly predicted as part of a given scope (*true positive*, further TP);
- How many were incorrectly predicted to be part of the scope (*false positive*, further FP);
- How many were incorrectly not predicted to be part of the scope (*false negative*, further FN).

Every token belonging to more than one instance is considered as one in each instance. For the *scope* level, all tokens of the instance must be predicted correctly in order to count the prediction as correct.

In the following example, there is one gold multi-word cue `CUE1` *no ... nor*:

	There	are	*no*	footsteps	*nor*	any	clue	to	the	criminals	.
GOLD			`CUE1`		`CUE1`						
PRED			`CUE1`				`CUE2`				

Token-level: TP - 1, FN - 1, FP - 1
Scope-level: TP - 0, FN - 1, FP - 2

On the token level, the token *no* is counted as a TP, and *nor* is counted as a FN. `CUE2` *any* that is incorrectly predicted will count towards token-level FPs. On the scope level, `CUE1` is only considered as a whole *no ... nor*. Since only *no* is predicted for `CUE1`, `CUE1` counts as a FN (as it was not fully predicted) and a FP (as what is predicted is not fully correct). Finally, `CUE2` *any* will count towards FPs, just as on the token level.

The scope-level event and scope metrics also depend on the correct prediction of the cue they are attached to. The scores can be calculated with a *full cue match*, which means that in the case of a multi-word cue, all parts of the cue should be predicted correctly. A score with *a partial match* allows only one token of the cue to be predicted in order to count the event and scope instances as correct. Finally, a score with *no cue match* is completely independent of the cue prediction.

	There	*are*	no	*footsteps*	nor	*any*	*clue*	*to*	*the*	*criminals*	.
GOLD	S	S	CUE1	E	CUE1	S	E	S	S	S	
PRED1	S	S	CUE1	E		S	E	S	S	S	
PRED2			CUE1	E	CUE1	S	E				

	PRED1	PRED2
Scope (token-level, full cue match):	TP - 0, FN - 8, FP - 8	TP - 3, FN - 5, FP - 0
Scope (token-level, partial cue match):	TP - 8, FN - 0, FP - 0	TP - 3, FN - 5, FP - 0
Scope (token-level, no cue match):	TP - 8, FN - 0, FP - 0	TP - 3, FN - 5, FP - 0

The example above illustrates two cases of multi-word cue prediction: `PRED2` has the cue fully predicted whereas `PRED1` failed to predict the second part (*nor*) of the cue. For a *full cue match* metric, the predicted scope `S` of `PRED1` would count both as a FN and a FP (scope-level) as even though the scope completely macthes the gold reference, the cue is underpredicted. For *partial cue match*, however, this would count as a TP, since at least one part of the predicted cue overlaps with the gold cue. On the other hand, `PRED2` has a full cue match, so, even though the scope is a FN / FP on scope-level as it is underpredicted, all of the correctly predicted tokens would count as

TPs on token-level for both *full* and *partial cue match*. As *no cue match* does not require any cue overlap, it would have the same scores as *partial match* here (and the scores would stay the same even if the cue would be predicted completely incorrectly). Note that events `E` are considered to be part of the scope.

Finally, the strictest metric is full negation which requires all parts of the cue and its scope (including events) to be correct. Both not predicting a gold part of the negation instance and predicting a part that is not in gold would discard the whole instance as a FN / FP. In the above examples, not one of the instances could be counted as TP for this metric as all of them had some parts predicted incorrectly.

As our base, we use the evaluation script of *SEM 2012 Shared Task (Morante and Blanco, 2012). The script includes several of the aforementioned metric types:

- Scope-level cue;
- Scope-level scope with both full cue match and partial cue match;
- Token-level scope with partial cue match;
- Scope-level event with no cue match;
- Full negation;
- Percentage of sentences that had fully correct predictions;
- Precentage of sentences with negation that had fully correct predictions.

In order to not punish the same faulty prediction twice, predictions that appear in the gold data but are not fully correct, are only counted as *false negatives* (these metrics will further be referred to as A-scores). As a follow-up request, Morante and Blanco (2012) also included scores that counted the mentioned cases as both *false negatives* and *false positives* (B-scores). The *SEM 2012 metrics also exclude punctuation from their evaluation.

The *SEM 2012 evaluation script is written in Perl and is specifically tailored to be used with Unix-formatted documents. Unfortunately, under

closer inspection, a bug in event evaluation was discovered, such that some of the false negatives were omitted, and some true positives were counted as false positives. The script also lacks the token-level metrics used in other works, providing only one for scope. Moreover, many previous works have used their own scripts and metrics to evaluate their systems, making it a challenge to compare between them. In order to cover as many metrics as possible as well as to provide a Python alternative to the Perl script and fix the event metrics, we re-implement the *SEM 2012 script in Python 3.7 and add some new metrics to the implementation. The new script includes:

- Token- and scope-level cue;
- Token- and scope-level scope with full cue match, partial cue match and no cue match;
- Token- and scope-level event with full cue match, partial cue match and no cue match;
- Full negation;
- Token-level "binary label" score (for cue, scope and event) that counts every token only once, even if it belongs to more than one scope;
- Percentage of sentences that had fully correct predictions;
- Precentage of sentences with negation that had fully correct predictions.

Scores for scope, full negation and sentence percentage have a version that excludes punctuation. For scope-level scores, both A-scores and B-scores are included.

Here, we report token- and scope-level cue, scope (partial match, no punctuation), event (partial match) and full negation (no punctuation). Scope-level scores are A-scores (i.e. punishing partially correct predictions only as false negative).

Dataset			Train	Dev	Test	Total
ConanDoyle-neg (reannotated)		sentence #	3,644	787	1,089	5,520
		sentence %	66%	14.3%	19.7%	100 %
		negation instance #	984	173	264	1,227
		negation sentence #	848	144	235	1,421
		negation sentence %	23.3%	18.3%	21.6%	25.7 %
		negation instance #	987	176	269	1,432
BioScope	Abstracts	sentence #	9,500	1,185	1,186	11,871
		sentence %	80%	10%	10%	100%
		negation instance #	1,396	156	167	1,719
		negation sentence #	1,297	148	152	1,597
		negation sentence %	13.7%	12.5%	12.8%	13.5%
	Full papers	sentence #	2,136	267	268	2,670
		sentence %	80%	10%	10%	100%
		negation instance #	293	45	38	376
		negation sentence #	268	41	30	339
		negation sentence %	12.5%	15.4%	11.2%	12.7%
SFU Review		sentence #	13,614	1,817	1,800	17,231
		sentence %	79%	11%	10%	100%
		negation instance #	2,835	365	309	3,509
		negation sentence #	2,503	328	276	3,107
		negation sentence %	18.4%	18.1%	15.3%	18%

Table 2: Dataset splits.

5.2 Experimental Setup

We use the official data split for *ConanDoyle-neg*, a 79-11-10 split for *SFU Review* (split by documents rather than sentences) and an 80-10-10 split for *BioScope* (see Appendix A for detailed information on the dataset splits). During the process of conversion to the dependency parsing format, we filter out 32 sentences with annotation errors or with a scope that belongs to a cue from a different sentence from *SFU Review*. Table 2 provides an overview of the data splits, as well as information on the negation instances in the splits (after filtering out). Negation sentence % is calculated as the number

of negation sentences over the number of sentences in the given split. We also create a version of *ConanDoyle-neg* with 10 multi-word cues reannotated as one-word cues (3 in training, 3 in development and 4 in test sets), specifically the cues *neither ... nor [... nor]*, *no ... nor* and *not ... not*. For example:

> *... **neither** Mr. Warren, **nor** I, **nor** the girl has once set eyes upon him.*

The example is annotated to have one multi-word cue in the original dataset. On the other hand, the reannotated version (further referred to as *ConanDoyle-neg* (reannotated)) treats every token of the cue as a separate instance, with every agent of the sentence (*Mr. Warren*, *I*, *the girl*) being a part of a scope of a separate instance, and the shared action (*has once set eyes upon him*) being an overlapping scope of the three instances. We use the same splits for the reannotated version as for the original one.

We train our models without a pre-defined number of epochs but using early stopping with the patience of 15 epochs. We use XLM-R as our transformer model. We optimize the model with an AdamW optimizer with a learning rate of $3e{-}5$. For a more detailed description of hyperparameters, refer to Appendix B. We report an average F1-score of 5 runs for every model.

We use NegBERT (Khandelwal and Sawant, 2020; Britto and Khandelwal, 2020) as our baseline. In order for the results to be comparable, we run NegBERT with our data splits and report the results obtained from our own runs and our own evaluation script. We use two embedding models: the multilingual XLM-R we use in STEPS (NegBERT$_{XLM\text{-}R}$), and English-based RoBERTa (NegBERT$_R$), one of the models used in the original implementation with the closest architecture to XLM-R. We use the hyperparameters reported in the original paper and train the model for 60 epochs, with an early stopping patience of 6 and a batch size of 8. As an optimizer, Adam with a learning rate of $3e^{-5}$ is used. We use the "augment" method for the cue encoding and the "average" method for the label post-processing as they are

reported to perform best (for more details on the methods, see Section 2.3). As event resolution was not performed by Britto and Khandelwal (2020), we add an additional event model based on NegBERT's scope model for *ConanDoyle-neg*. The results are averaged over 5 runs. We believe the original results from the papers for scope resolution were obtained by feeding gold cues to the system, so we report them as well (further referred to as NegBERT (gold cues), see Appendix C). However, we believe the results are not comparable to our models as we only feed the tokens to the parser, not the gold cues, so we do not use these results as a baseline.

NegBERT outputs cue detection results as a list of labels, where 0 corresponds to affixational cue, 1 to one-word cue and 2 to multi-word cue. The scope and event predictions are outputted in the same fashion, separately for every cue as a binary list of labels. For evaluation, we convert the predictions into *SEM 2012 format. An issue arises when converting multi-word cues, as there can be more than one multi-word cue in one sentence, but NegBERT's labeling scheme does not add any means of differentiating between them. For example:

[No], [no], my dear Watson, [not at all], [by no means all].

The brackets confine every separate cue in the example. As these kinds of cases are not very frequent, we do not implement any algorithm for checking for such cues, and write all of the tokens labeled as 2 into one cue (i.e. in the given example, the converter would write *[not at all ... by no means all]* as one cue). This is more typical for SFU Review, which annotates cases like *didn't* as one cue consisting of two tokens, i.e. a multi-word cue, for example:

If you buy it [do n't] say I [did n't] warn you.

We also compare our results to those reported by Kurtz et al. (2020) for experiments on *ConanDoyle-neg*. Their scores were obtained using the official *SEM 2012 evaluation script where the event was meant to be measured with

no cue match. As the results for the event score we report are based on partial cue match, and since there was a bug related to the score found in the script, we believe their event score is not directly comparable. However, we still include it in our table for completeness. Furthermore, we only compare to Kurtz et al. (2020) in the experiments on the original *ConanDoyle-neg* as that is the dataset they used.

5.3 Experiment set 1: Exploring Dependency Mappings

In our first experiment set, we aim to explore different mappings and settings to find the best performing ones. We use the original *ConanDoyle-neg* for this purpose. Out of our five mappings, the nested mapping yields the best results for every score except scope-level cue in our base settings (see Table 3). Syntactic mapping provides the best result on token-level and second best result on scope-level for cue detection, span-like mapping for event resolution, and finally direct mapping for scope resolution and the full score corresponding to an overall correctness of the whole instance.

To investigate the influence of the actual syntactic dependencies on the task, we test our mappings by replacing the intial XLM-R model with an XLM-R previously trained on *Universal Dependency Treebank* (Silveira et al., 2014) using STEPS (further referred to as syntactic XLM-R). As the boundaries of the scope of the given cue are syntactically grounded, we expected these settings to be the most influential for the scope resolution. For this experiment we use the direct mapping as our base one, the nested mapping as the best performing one, and the syntactic and the syntactic-direct as the syntax-based ones. The token-level cue score increases for the syntactic-direct mapping, and the scope-level cue score increases for all of the mappings except direct in these settings. This indicates that syntactic XLM-R was especially helpful for prediction of the multi-word cues. Scope and event scores increase for every mapping, having the greatest impact on the syntactic and syntactic-direct mapping. The token-level scope score of the syntactic

	Cue		Scope		Event		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Token*	*Scope*	*Scope*
Kurtz et al. (2020)	-	92.68	87.91	-	-	*63.69	59.40
NegBERT$_R$	91.25	90.82	86.88	73.26	68.29	69.95	53.30
NegBERT$_{XLM\text{-}R}$	89.61	89.28	86.31	75.67	67.03	69.02	53.54
Direct mapping	91.57	92.20	86.82	74.31	65.42	66.24	57.78
Nested mapping	**92.78**	92.98	*87.54*	*74.71*	*68.12*	*68.32*	*58.69*
Span-like mapping	92.56	92.91	79.01	70.07	66.59	67.54	56.08
Syntactic mapping	92.69	*93.11*	78.94	64.18	65.23	65.64	50.23
Syntactic-direct mapping	92.21	92.54	80.31	67.53	64.65	65.92	52.57
Syntactic XLM-R							
Direct mapping	91.22	91.89	**88.53**	**79.82**	68.21	70.33	**66.19**
Nested mapping	*92.77*	93.03	87.90	79.68	**68.99**	**70.80**	66.17
Syntactic mapping	92.69	**93.28**	85.19	75.79	67.70	68.73	60.87
Syntactic-direct mapping	92.41	92.85	84.63	73.95	66.50	67.87	58.92
Multitask							
Direct mapping	91.05	91.88	87.28	73.03	66.22	66.60	54.97
Nested mapping	*92.06*	*92.69*	*87.43*	*74.56*	*67.45*	*68.23*	*58.34*
Syntactic mapping	91.98	92.56	85.06	73.29	66.73	67.34	57.18
Multi + Synt XLM-R							
Nested mapping	91.86	92.34	85.39	72.53	67.29	67.08	56.46

Table 3: Experiment results (F1-scores). Train / test set: *ConanDoyle-neg*. Best result for a given metric is in bold, best performing mapping in the same settings in italics.

mapping gains as much as 6.25%, and the scope-level score gains 11.61%, indicating that not only more scope tokens were detected correctly, but also that all tokens of the scope were found for more negation instances. The full negation score also increases significantly for every mapping. The lesser change of the scope scores in comparison to the full score for nested and direct mapping suggests that the usage of the syntactic transformer model contributes especially to collecting all of the tokens belonging to the scope of the same instance. However, the best event results out of all mappings were

again achieved using the nested mapping, with the direct mapping providing the best scope and full negation scores.

Another approach to incorporating syntax in our model is multitask training. We train our models in parallel on the negation data of *ConanDoyle-neg* and the enhanced dependency data of the *Universal Dependency Treebank*. We use the mappings that performed best in the experiments with the syntactic XLM-R, namely direct, nested and syntactic. The multitask setting mostly influences the scope metrics of the syntactic mapping, making the performance 6.2% higher on token-level and 9.11% higher on scope-level, as well as the full negation score with a gain of 6.95% in the F1-score. The event score also increases for direct and syntactic mappings. The cue metrics, however, decrease for all of the mappings. Finally, multitask training has a negative impact on the nested mapping results, showing a decrease in all of the scores. Nevertheless, the nested mapping still performs best in the given settings. In general, using a pre-trained model appears to be much more effective for the task than training the model with negation and dependencies side by side.

Finally, we take our best performing mapping (nested) and train it in the same multitask settings, but on the syntactic XLM-R. This does not prove to be beneficial and decreases the results for the mapping even further. Thus, we conclude that the usage of syntax for negation detection is only fruitful in moderation.

Overall, the best results of this experiment set were achieved with the use of different settings and different mappings. With regard to the cue detection, the best token-level score was achieved using nested mapping in the base settings while the best scope-level score was achieved with a syntactic mapping with the use of the syntactic XLM-R. The XLM-R model seems to be helpful for detection of all tokens of the multi-word cues in this case. The syntactic model is helpful for scope and event resolution, yielding the best scores for scope and full instance with the direct mapping and for events with the nested mapping.

Most of our mappings perform on a par with or outperform Kurtz et al. (2020) and NegBERT with the exception of scope resolution and full negation detection, where three of our mappings (span-like, syntactic and syntactic-direct) perform significantly lower. However, both syntactic and syntactic-direct mappings outperform the two baselines when trained on a pre-terained syntactic XLM-R. The nested and direct representations with the syntactic XLM-R (the best performing mapping-settings combinations) outperform the baselines in event resolution and full negation detection. The direct mapping performs better for scope resolution while the nested mappings achieves better results for cue detection. In general, our systems mostly gain in performance in scope and event resolution.

5.4 Experiment set 2: *ConanDoyle-neg* (Reannotated)

We test the mappings in the base settings and with the usage of the syntactic XLM-R on *ConanDoyle-neg* (reannotated). We expect the models to perform better for the cue detection with this dataset, as some of the errors made by the models with the original dataset were related to the prediction of one multi-word cue, e.g. *neither ... nor*, as several one-word cues, i.e. *neither* and *nor* separately for the given example.

The cue scores increase for all mappings for this dataset (see Table 4). Direct mapping performs best for this dataset for scope and event resolution and full negation detection, with the nested mapping achieving the best token-level cue score and the syntactic mapping achieving best scope-level score. The best results overall are achieved by the same models when using XLM-R with an exception of scope-level scope metric performing slightly better for nested mapping.

Scores for all metrics increase for NegBERT$_R$ when trained and tested on reannotated *ConanDoyle-neg* except for the token-level events. On the other hand, there is a much greater F1-score increase for NegBERT$_{XLM\text{-}R}$, especially for full negation. Our best models trained on the syntatic XLM-R,

	Cue		Scope		Event		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Token*	*Scope*	*Scope*
NegBERT$_R$	91.52	91.57	87.87	77.62	67.88	69.98	53.63
NegBERT$_{XLM\text{-}R}$	91.76	91.84	87.88	78.83	68.19	**71.30**	60.01
Direct mapping	93.13	93.47	*87.24*	*73.19*	*66.49*	*66.25*	*57.77*
Nested mapping	*93.66*	93.95	85.65	72.80	65.39	64.75	55.94
Syntactic mapping	93.61	93.86	79.28	65.50	65.35	66.01	51.97
Syntactic-direct mapping	93.62	*93.97*	80.80	65.58	65.24	66.24	52.54
Syntactic XLM-R							
Direct mapping	93.32	93.61	**88.03**	79.05	**68.40**	*70.63*	**65.88**
Nested mapping	**93.89**	94.10	87.21	**79.18**	67.45	70.05	65.54
Syntactic mapping	93.44	93.82	84.94	75.64	67.04	68.34	61.14
Syntactic-direct mapping	93.83	**94.19**	84.87	74.27	65.65	67.01	59.16

Table 4: Experiment results (F1-scores). Train / test set: *ConanDoyle-neg* (reannotated).

while being quite close to the NegBERT$_{XLM\text{-}R}$ results, mostly outperform them. However, NegBERT$_{XLM\text{-}R}$ achieves a greater scope-level event score. The results differ most notably in full negation, which suggests that the NegBERT model works better on the token level, while our dependency-parsing based models are able to capture all parts of the same negation instance more often.

During the experiments, we have observed training instability, with different models of the same mapping having a high performance range, especially for full negation and scope on scope-level. For score distribution visualization, refer to Appendix D. For the scope-level scope metric, the results range[1] as much as 10.84% for the direct mapping, 12.38% for the nested mapping, 6.38% for the syntactic mapping and 8.81% for the syntactic-direct mapping. The nested mapping is the most susceptible to the issue, with the a range of 1.15% for scope-level cue metric, 10.2% for scope-level event metric and

[1]Range here refers to the difference between the minimum and maximum scores.

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: BioScope Abstracts				
	Train: *BioScope Abstracts*				
NegBERT$_R$	92.33	90.47	86.14	85.75	83.00
NegBERT$_{XLM\text{-}R}$	92.44	91.45	87.04	86.44	83.19
Direct mapping	97.08	**96.42**	89.65	83.90	82.58
Nested mapping	97.00	96.15	90.62	87.38	86.36
Syntactic XLM-R					
Direct mapping	97.05	96.12	91.49	88.19	87.90
Nested mapping	**97.11**	96.25	**92.40**	**89.08**	**88.74**
	Train: *BioScope Full Papers*				
NegBERT$_R$	92.26	91.60	**86.00**	81.25	78.57
NegBERT$_{XLM\text{-}R}$	91.64	89.68	84.19	**81.89**	**79.40**
Direct mapping	94.23	94.12	78.83	58.86	56.48
Nested mapping	**94.63**	**94.53**	79.65	66.29	64.69
Syntactic XLM-R					
Direct mapping	94.53	94.37	*81.34*	*72.73*	*72.24*
Nested mapping	94.10	93.92	80.05	66.74	66.38

Table 5: Experiment results (F1-scores). Train set: *BioScope*. Test set: *BioScope Abstracts*.

13.92% for the full negation metric. Using syntactic XLM-R significantly reduces the issue for the scope, event and full negation metrics, with nested mapping only having a range of 2.21% for scope-level scope metric, 4.6% for event metric and 2.49% for the full negation metric. The use of the syntactic XLM-R has has little to no effect on the syntactic-direct mapping; however, the syntactic and the syntactic-direct mapping appear to be least susceptible to the issue in general.

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: BioScope Full Papers				
	Train: *BioScope Full Papers*				
NegBERT$_R$	83.25	82.00	**71.60**	64.43	64.43
NegBERT$_{XLM\text{-}R}$	83.14	81.81	69.12	**71.47**	**70.44**
Direct mapping	83.86	83.46	55.51	36.30	34.34
Nested mapping	**84.11**	**83.71**	55.83	*43.33*	*41.90*
Syntactic XLM-R					
Direct mapping	82.76	82.34	55.71	43.06	41.48
Nested mapping	83.41	83.01	*57.25*	37.69	36.22
	Train: *BioScope Abstracts*				
NegBERT$_R$	80.15	79.61	69.26	65.02	**64.01**
NegBERT$_{XLM\text{-}R}$	79.91	78.97	66.72	64.03	61.29
Direct mapping	**86.97**	**86.65**	69.32	58.62	55.29
Nested mapping	86.41	86.07	70.43	62.55	61.21
Syntactic XLM-R					
Direct mapping	86.44	86.09	73.63	63.43	62.60
Nested mapping	86.71	86.37	**75.13**	**65.35**	*63.99*

Table 6: Experiment results (F1-scores). Train set: *BioScope*. Test set: *BioScope Full Papers*.

5.5 Experiment set 3: Other Domains

To evaluate our models on the two other datasets, *BioScope* and *SFU Review*, we only use direct and nested mappings as our best performing mappings. We test them with the syntactic XLM-R as well.

For *BioScope* we do not concatenate *Abstracts* and *Full Papers* and train and test the models separately on each of the subsets. The models trained on *BioScope Abstracts* perform best when tested on both *Abstracts* (see Table 5) and *Full Papers* (see Table 6). We believe the reason for that is the

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
NegBERT$_R$	81.60	80.92	73.63	74.34	72.51
NegBERT$_{XLM\text{-}R}$	82.47	81.85	74.50	74.91	73.60
Direct mapping	84.43	86.66	76.64	73.98	73.11
Nested mapping	84.50	86.53	76.29	73.33	72.35
Syntactic XLM-R					
Direct mapping	84.85	**87.03**	77.00	74.83	74.12
Nested mapping	**85.03**	86.99	**78.10**	**76.86**	**75.99**

Table 7: Experiment results (F1-scores). Train / test set: *SFU Review*.

difference in size of the two subsets, with *Abstracts* being over 4 times larger than *Full Papers*. The model trained on *Abstracts* data encoded with nested mapping performs best for scope resolution and full negation detection, with a performance gain when used with a syntactic XLM-R. For cue detection, the scores do not differ much between the models, with best predictions with the nested mapping and a syntactic XLM-R on cue token level and the direct mapping on scope level for making prediction for *Abstracts*, and the nested mapping on both levels for *Full Papers*. When trained on *Full Papers*, the nested mapping performs best when testing on the same dataset, with the syntactic XLM-R version performing better for the scope resolution on the token level. For predictions on *Abstracts*, the nested mapping with syntactic XLM-R generally performs best, with the cue metrics being slightly higher for direct mapping in base settings.

NegBERT models perform better for the scope resolution and full negation detection than STEPS when trained on *Full Papers*, with NegBERT$_{XLM\text{-}R}$ performing especially well for the scope-level metrics. This is indicative of the greater dependence on the amount of data for STEPS in comparison to NegBERT. The NegBERT$_R$ model also performs slightly better for the full negation detection when trained and tested on *Abstracts* than the STEPS model with the nested mapping and the syntactic XLM-R.

For *SFU Review*, the nested mapping model trained on syntactic XLM-R performs best for all scores (see Table 7), with the direct mapping trained on syntactic XLM-R performing slightly better for cue predictions on scope level. The best STEPS model strongly outperforms both of the NegBERT models.

5.6 Experiment set 4: Cross-Domain

We perform cross-domain testing of the models from the previous experiments (for *ConanDoyle-neg* we use the models trained on the reannotated version). For a full F1-score report of the results, see Appendix E.

The model trained on *SFU Review* is able to predict best for *ConanDoyle-neg* (reannotated), with the highest cue detection results achieving 76.36% and 73.84% F1-score on the token and scope levels, respectively; the highest scope detection results 67.2% and 15.04% on the token and scope levels, respectively; and the highest full negation results 15.32% (see Table 14 of Appendix E). The results were achieved with a direct mapping model, with scope results using a model trained on syntactic XLM-R. All of the models, both STEPS and NegBERT (regardless of the training dataset), show very low scope-level scope and full negation scores in cross-domain settings. This implies that even though the models are able to predict some parts of the scopes, they fail to predict the full scope of an instance. This is most likely due to different scope boundary annotation in the datasets.

The *BioScope* datasets achieve the best predictions in cross-domain settings with the model trained on *SFU Review* as well (see Tables 15 and 16 of Appendix E). This result is explainable by the fact that the *SFU Review* annotation guidelines built on the *BioScope* annotation guidelines, thus, the annotation scheme of *BioScope* and *SFU Review* are much closer to each other than to that of *ConanDoyle-neg*. The model using the nested mapping and the syntactic XLM-R yields the overall best results, with the direct mapping model with the syntactic XLM-R performing best for the scope-

level cue detection for *BioScope Full Papers* and token-level scope detection for *BioScope Abstracts* and the nested mapping model in the base settings performing best for the cue detection for *BioScope Abstracts*.

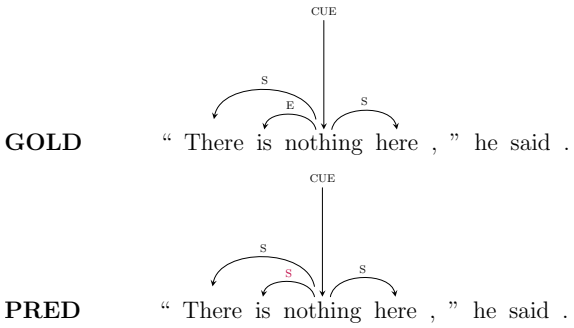
The *SFU Review* corpus is best predicted with the model trained on *BioScope*, with *Abstracts* being first and *Full Papers* being second best (see Table 17 of Appendix E). It should be noted that *Full Papers*, despite its size, performs better than *ConanDoyle-neg* due to closer annotation schemes with *SFU Review*. The direct mapping model performs best, with the one using the syntactic XLM-R performing better for scope and token-level cue detection.

Both of the NegBERT models tend to outperform the STEPS models in cross-domain experiments when tested on *ConanDoyle-neg* (reannotated) and *SFU Review*, indicating that the dependency-based models are more domain- and annotation-scheme dependent, and NegBERT models are more generalizable. The NegBERT models also outperform STEPS when trained on *ConanDoyle-neg* and tested on *BioScope Full Papers*. As for the rest of the experiments on *BioScope*, STEPS generally yields better results. Both NegBERT and STEPS achieve the best results when trained and tested on the same dataset, indicating that all in all, generalization remains a widespread challenge for both dependency-parsing based and sequence-tagging based approaches.

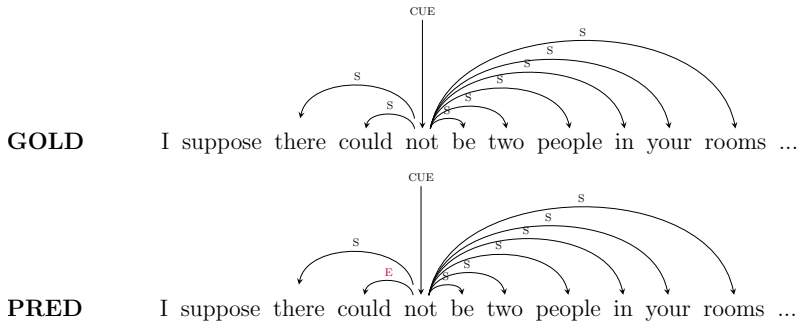
6 Discussion

6.1 Error Analysis

For *ConanDoyle-neg*, the models achieve the lowest scores for the event prediction among cue, scope and event detection. The reason behind this is the annotation scheme. The annotation guidelines of *ConanDoyle-neg* state that the event is to be annotated only if it is factual (Morante et al., 2011; p. 32). However, factuality is a complex concept which the models do not easily learn. This leads to many FP predictions of the event when it is non-factual (see Figure 13b) and the frequent FN abscence of predictions of the event when it is factual (see Figure 13a). This is the most frequent error that the models trained on *ConanDoyle-neg* make.



(a) An incorrectly not predicted event (FN).



(b) An incorrectly predicted event (FP).
A modal construction is not marked as an event.

Figure 13: Incorrect event predictions (*ConanDoyle-neg*).

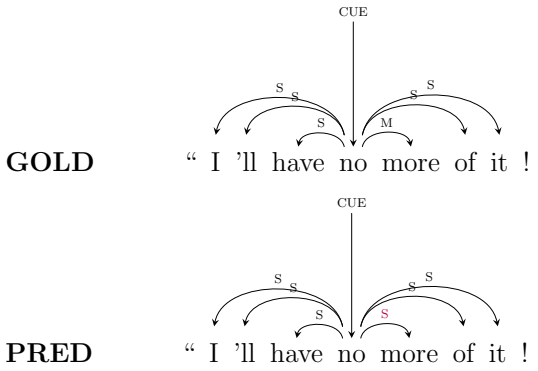
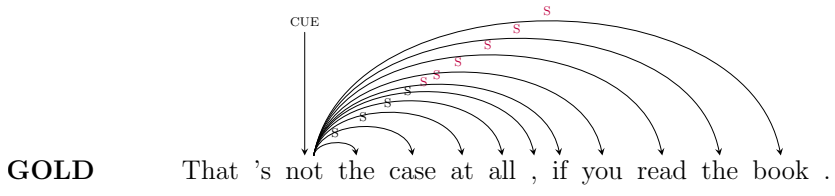


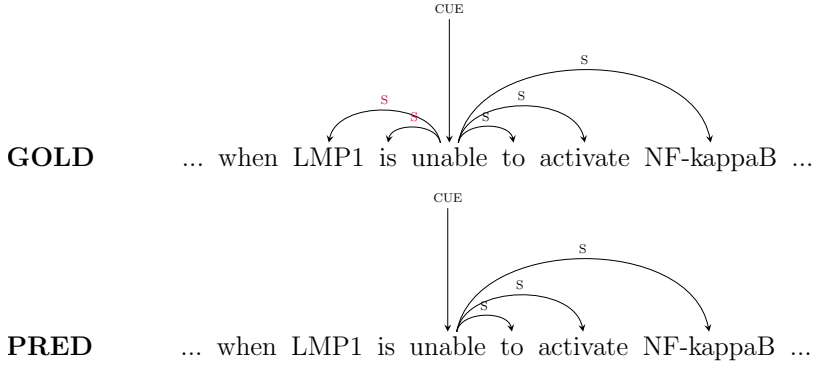
Figure 14: Partially predicted multi-word cue.

Another source of prediction errors is multi-word cues such as *no more*, *never more* and *absolutely nothing*. While the model is able to label the most obvious negation part as the cue (i.e. *no*, *never*, *nothing*), it fails to detect the rest of the cue (see Figure 14). The reason for this might be the lack of data that includes such cues: the training set does not have either of the listed cues, but contains other multi-word cues instead. A less frequent but occurent cue-related issue is the prediction of affixational cues. The models sometimes label as cues the words that look similar to the affixational cues but are not actually cues (e.g. *innocent*). Moreover, some of the words that can be an affixational cue in a certain context might not serve as such in a different context but still be predicted as such (e.g. *You are uneasy ...*).

The scope-level scope metric is affected by some tokens of a scope not being predicted as such. For all three datasets, conjunctions and prepositions such as *if*, *with*, *except*, *and*, *but*, as well as commas in some cases play the role of scope delimiters, with some of the scope tokens being “cut-off” when occuring after such a delimiter or when embedded within commas (see Figure 15a). The models trained on *BioScope* also occasionally fail to predict the tokens of the scope that are located in front of the cue (see Figure 15b). The models trained on syntactic XLM-R handle these kinds of cases better, most probably because they are based on the syntactic relations of the sentence rather than just surrounding lexical infromation.



(a) Scope cut off by delimiters "*, if*" (*SFU Review*).



(b) Scope without the tokens in front of the cue predicted (*BioScope*).

Figure 15: Partially predicted scopes.

The evaluation results for *SFU Review* generally show a lower precision in comparison to the recall. This is due to many negated constructions being predicted as such, but not annotated as such. For example, none of the following sentences were annotated as negated but were predicted as negated:

... it didn't really speak to me.

I really don't have many complaints ...

If you can't afford a whole set ...

For the original version of *ConanDoyle-neg*, many mappings fail to predict multi-word cues such as *neither...nor*. The span-like mapping models predict

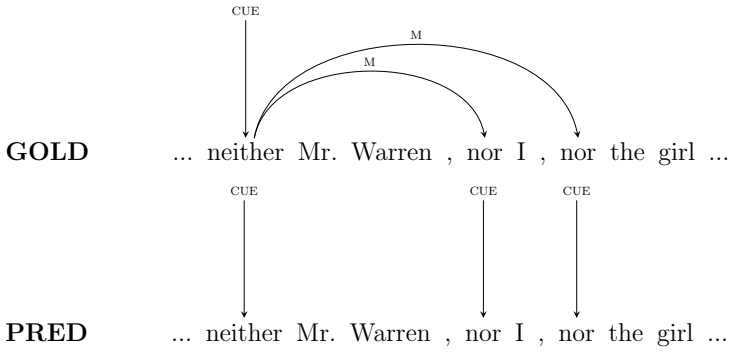


Figure 16: Multi-word cue prediction with span-like mapping (*ConanDoyle-neg* (orig.))).

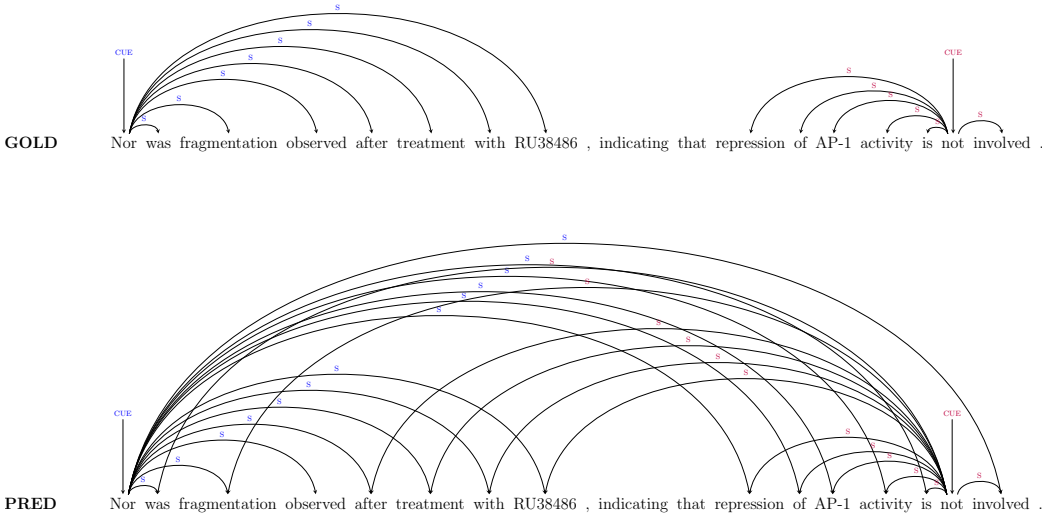


Figure 17: Direct mapping scope prediction for the two cues in the same sentence (*BioScope*).

each part of the cue as a separate one-word cue (see Figure 16; note that for simplicity, only cues are shown in the figure; scopes and events are omitted). The nested, syntactic and syntactic-direct mappings tend to predict the first token of the cue as the cue, with the rest of the token predicted as a part of the scope.

The direct mapping models have a tendency to link most of the tokens

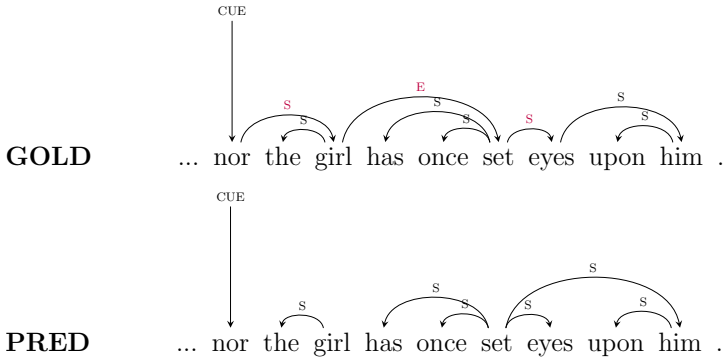


Figure 18: Prediction of a syntactic mapping model (*ConanDoyle-neg* (re-ann.)).

labeled as scope to every cue in the sentence when there is more than one cue (see Figure 17). The nested mapping models avoid such errors, presumably, because they learn fewer cases with one token being a scope dependent of several cues. The use of the syntactic XLM-R also helps to reduce this problem, possibly due to its more sytactic reasoning for arc assignment.

An error specific to the syntactic mapping significantly hurts the models scope and event detection performance. The model is able to predict the scope and event arc but fails to make an arc from the cue to one of the scope or event tokens. This leads to completely disregarding the scope and event prediction as not belonging to a cue. In the example in Figure 18, despite most of the scope tokens being predicted, all of them are discarded as there is no arc coming from the cue. The syntactic-direct mapping is less prone to such errors, which is probably due to the fact that it learns more arcs coming out from the cue token than the syntactic mapping.

6.2 Outlook

We have explored five different ways to encode negation data as dependency graphs; however, we realize that there is still room for improvement of the suggested mappings. The encodings can be evolved further to avoid some

of the flaws discovered in this work. For example, simplifying the span-like mapping to create a more intuitive and more machine-learnable encoding may provide fruitful results. Moreover, developing more sophisticated post-processing tactics for some of the mappings can improve the performance of the corresponding models. For example, additional post-processing can be applied to the scope that was predicted without any cue to determine whether it should be linked to a cue or discarded. Furthermore, a different neural model can be developed; for example, a "hybrid" model combining a tagger for cue detection with a dependency parser for scope prediction, where the latter would be fed the predicted cues as the input.

One of the challenges of the task is the existence of a large number of different metrics used to evaluate negation detection with no universally accepted metric or definition of the way the existing metrics should be measured. With our implementation of an evaluation script, we hope to create an evaluation basis for future works and to make it possible to obtain scores comparable to the past works. Systematic comparison of other approaches applied previously using the same evaluation process would be a great contribution to the proper investigation of the phenomenon of negation.

Another challenge of the task that remains open is poor domain generalization, largely due to the annotation differences within each domains. A potential improvement step in that direction may include developing a semi-automatic means of annotation unification for the existing negation datasets.

7 Conclusion

In this thesis, we have approached negation resolution as a dependency parsing task by transforming the data into dependency graphs and applying a dependency parser to detect negation instances in text. Our aim was to investigate the influence of such an approach on a task not typical for parsing. We were guided by the idea of a close relation between the negation scope boundaries and the syntactic structure of the sentence. We have presented four novel linguistically motivated encodings for approaching the negation resolution task as dependency parsing and compared the encodings against each other, as well as against the encoding suggested by Kurtz et al. (2020), which inspired this work. We have found the nested encoding to perform best in most cases, especially for scope resolution and full negation detection, with the direct mapping sometimes performing better for cue detection. We have looked into the advantages and disadvantages of different encodings, as well as the reasons behind their different ways of behaving when used for the negation detection task.

We have tested our dependency-based approach against two sequence-labeling baseline models, with sequence labeling being the typical approach for the negation detection task. We have shown that our direct and nested mapping models with the use of the syntactic XLM-R outperform the sequence labeling approach on in-domain experiments, proving that syntax is a useful tool for identifying the scope boundaries. We have also investigated the flaws of our model, finding that its performance is rather dependent on the size of the data. We have performed cross-domain experiments, where the sequence tagging models have outperformed our models in cases when the training and test datasets differed significantly in annotation. Overall, domain generalization has proven to still be a challenge.

Finally, we have implemented our own evaluation script for a range of existing metrics for negation detection to make a fair comparison between the different systems possible, to collect the greatly varying metrics into one

script and establish an evaluation basis for future works.

References

Jorge Carrillo de Albornoz, Laura Plaza, Alberto Díaz, and Miguel Ballesteros. 2012. UCM-I: A rule-based syntactic approach for resolving the scope of negation. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 282–287, Montréal, Canada. Association for Computational Linguistics.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas. Association for Computational Linguistics.

Benita Kathleen Britto and Aditya Khandelwal. 2020. Resolving the scope of speculation and negation using transformer-based architectures.

W. Chapman, W. Bridewell, P. Hanbury, G. Cooper, and B. Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34 5:301–10.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Isaac Councill, Ryan McDonald, and Leonid Velikovich. 2010. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 51–59, Uppsala, Sweden. University of Antwerp.

Noa Cruz Diaz, Maite Taboada, and Ruslan Mitkov. 2015. A machine learning approach to negation and speculation detection for sentiment analysis. *Journal of the American Society for Information Science and Technology (JASIST)*.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 495–504, Berlin, Germany. Association for Computational Linguistics.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2018. Neural networks for cross-lingual negation scope detection.

Federico Fancellu and Bonnie Webber. 2015. Translating negation: Induction, search and model errors. In *Proceedings of the Ninth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 21–29, Denver, Colorado, USA. Association for Computational Linguistics.

Stefan Grünewald, Annemarie Friedrich, and Jonas Kuhn. 2021. Applying occam's razor to transformer-based dependency parsing: What works, what doesn't, and what is really necessary.

Dan Jurafsky and James H. Martin. 2020. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Third Edition draft)*. Draft of December 30, 2020.

Aditya Khandelwal and Benita Kathleen Britto. 2020. Multitask learning of negation and speculation using transformers. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 79–87, Online. Association for Computational Linguistics.

Aditya Khandelwal and Suraj Sawant. 2020. NegBERT: A transfer learning approach for negation detection and scope resolution. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5739–5748, Marseille, France. European Language Resources Association.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9:10.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Natalia Konstantinova, Sheila C.M. de Sousa, Noa P. Cruz, Manuel J. Maña, Maite Taboada, and Ruslan Mitkov. 2012. A review corpus annotated for negation, speculation and their scope. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3190–3195, Istanbul, Turkey. European Language Resources Association (ELRA).

Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO 2: Sequence-labeling negation using dependency features. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 319–327, Montréal, Canada. Association for Computational Linguistics.

Junhui Li, Guodong Zhou, Hongling Wang, and Qiaoming Zhu. 2010. Learning the scope of negation via shallow semantic parsing. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 671–679, Beijing, China. Coling 2010 Organizing Committee.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Nick McKenna and Mark Steedman. 2020. Learning negation scope from syntactic structure. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 137–142, Barcelona, Spain (Online). Association for Computational Linguistics.

Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C Max Schmidt, Hongfang Liu, and Mathew Palakal. 2015. Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, 54:213—219.

Andrew Moore and Jeremy Barnes. 2021. Multi-task learning of negation and speculation for targeted sentiment classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2838–2869, Online. Association for Computational Linguistics.

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 shared task: Resolving the scope and focus of negation. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 265–274, Montréal, Canada. Association for Computational Linguistics.

Roser Morante and Walter Daelemans. 2009. A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 21–29, Boulder, Colorado. Association for Computational Linguistics.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in conan doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1563–1568, Istanbul, Turkey. European Language Resources Association (ELRA).

Roser Morante, Sarah Schrauwen, and Walter Daelemans. 2011. Annotation of negation cues and their scope. Guidelines v1.0. Technical report, CliPS, University of Antwerp, Antwerp.

Joakim Nivre. 2005. Dependency grammar and dependency parsing. Technical Report MSI 05133, Växjö University, School of Mathematics and Systems Engineering.

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. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc.

Romain Paulus, R. Socher, and Christopher D. Manning. 2014. Global belief recursive neural networks. In *NIPS*.

Yifan Peng, Xiaosong Wang, Le Lu, Mohammadhadi Bagheri, Ronald Summers, and Zhiyong Lu. 2017. Negbio: a high-performance tool for negation and uncertainty detection in radiology reports.

Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. UiO1: Constituent-based discriminative ranking for negation resolution. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 310–318, Montréal, Canada. Association for Computational Linguistics.

L. Rokach, Roni Romano, and O. Maimon. 2008. Negation recognition in medical narrative reports. *Information Retrieval*, 11:499–538.

Olivia Sanchez Graillet and Massimo Poesio. 2007. Negation of protein protein interactions: Analysis and extraction. *Bioinformatics (Oxford, England)*, 23:i424–32.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sunghwan Sohn, Stephen Wu, and Christopher Chute. 2012. Dependency parser-based negation detection in clinical narratives. *AMIA Summits on Translational Science proceedings AMIA Summit on Translational Science*, 2012:1–8.

György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45, Columbus, Ohio. Association for Computational Linguistics.

Maite Taboada, Caroline Anthony, and Kimberly Voll. 2006. Methods for creating semantic orientation dictionaries. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

James Paul White. 2012. UWashington: Negation resolution using machine learning methods. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 335–339, Montréal, Canada. Association for Computational Linguistics.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 60–68, Uppsala, Sweden. University of Antwerp.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

A Split IDs

Table 8 provides information on the sentences and documents that we used for the data splits of *BioScope* and *SFU Review* for our experiments.

Dataset		**Sentence IDs**
Bioscope Abstracts	train	S1 - S1017
	dev	S1018 - S1147
	test	S1148 - S1273
Bioscope Full Papers	train	S1 - S6; S7.1 - S7.7
	dev	S7.8 - S7.274
	test	S7.275 - S7.330; S8 - S9
Dataset		**File names**
SFU Review	train	yes[1-20].xml, no[1-20].xml
	dev	yes[21-22].xml, no[21-23].xml
	test	yes[23-25].xml, no[24-25].xml

Table 8: Sentences IDs (*BioScope*) and file names (*SFU Review*) for sentences from the data splits used in the experiments.

B Hyperparameters

Table 9 lists the hyperparameters used with the STEPS models.

Transformer LM	
Token mask probability	0.15
Layer dropout	0.1
Hidden dropout	0.2
Attention dropout	0.2
Output dropout	0.5
Biaffine classifier	
Arc and label scorer dimension	1024
Dropout	0.33
Optimization	
Optimizer	AdamW
Weight decay	0
Batch size	32
Base learning rate	$4e^{-5}$
LR schedule	Noam
LR warmup	1 epoch

Table 9: Hyperparameter values used for STEPS.

C NegBERT Results

The following tables report results for the NegBERT models (Khandelwal and Sawant, 2020; Britto and Khandelwal, 2020) trained on various datasets and tested on a specific dataset. Table 10 reports results for testing on *ConanDoyle-neg*, both original and reannotated, Table 11 for testing on *BioScope Abstracts*, Table 12 for testing on *Full Papers* and Table 13 for testing on *SFU Review*. NegBERT$_R$ refers to the NegBERT model trained on RoBERTa and NegBERT$_{XLM\text{-}R}$ to the model trained on XLM-R. The results are reported for two versions of the scope and event prediction:

(a) NegBERT: the model was fed predicted cues;
(b) NegBERT (gold cues): the model that was fed gold cues.

	Cue		**Scope**		**Event**		**Full**
Level	*Token*	*Scope*	*Token*	*Scope*	*Token*	*Scope*	*Scope*
			Test: *ConanDoyle-neg*				
			Train: *ConanDoyle-neg*				
NegBERT$_R$	**91.25**	**90.82**	86.88	73.26	68.29	69.95	53.30
NegBERT$_{XLM\text{-}R}$	89.61	89.28	86.31	75.67	67.03	69.02	53.54
NegBERT$_R$ (gold cues)	-	-	91.05	79.36	**71.87**	**74.72**	59.06
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	**91.78**	**83.18**	71.69	**74.72**	**60.27**
			Test: *ConanDoyle-neg* (reannotated)				
			Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	91.52	91.57	87.87	77.62	67.88	69.98	53.63
NegBERT$_{XLM\text{-}R}$	**91.76**	**91.84**	87.88	78.83	68.19	71.30	60.01
NegBERT$_R$ (gold cues)	-	-	**92.73**	84.59	**73.05**	75.84	59.80
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	92.55	**85.39**	72.40	**76.18**	**65.59**
			Train: *BioScope Abstracts*				
NegBERT$_R$	*77.10*	*74.52*	65.82	13.85	-	-	11.02
NegBERT$_{XLM\text{-}R}$	71.40	67.72	61.70	13.03	-	-	13.38
NegBERT$_R$ (gold cues)	-	-	76.20	19.26	-	-	16.09
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	*76.54*	*19.38*	-	-	*17.51*
			Train: *BioScope Full Papers*				
NegBERT$_R$	*76.09*	*74.66*	64.39	14.41	-	-	11.55
NegBERT$_{XLM\text{-}R}$	68.91	68.90	58.86	12.97	-	-	8.77
NegBERT$_R$ (gold cues)	-	-	*75.73*	*20.49*	-	-	*16.59*
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	74.74	20.35	-	-	14.20
			Train: *SFU Review*				
NegBERT$_R$	77.07	*74.73*	69.23	14.55	-	-	14.97
NegBERT$_{XLM\text{-}R}$	*77.30*	74.38	68.24	15.25	-	-	14.67
NegBERT$_R$ (gold cues)	-	-	*74.97*	19.16	-	-	18.97
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	73.75	*19.96*	-	-	*19.83*

Table 10: Experiment results for NegBERT (F1-scores). Test set: *ConanDoyle-neg* (original / reannotated).

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: *BioScope Abstracts*				
	Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	60.66	58.87	51.88	24.32	20.24
NegBERT$_{XLM\text{-}R}$	*63.40*	*61.66*	52.93	24.53	21.84
NegBERT$_R$ (gold cues)	-	-	*79.31*	*46.62*	40.93
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	76.54	44.94	*40.99*
	Train: *BioScope Abstracts*				
NegBERT$_R$	92.33	90.47	86.14	85.75	83.00
NegBERT$_{XLM\text{-}R}$	**92.44**	*91.45*	87.04	86.44	83.19
NegBERT$_R$ (gold cues)	-	-	**94.79**	**93.51**	**91.97**
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	93.60	92.65	91.55
	Train: *BioScope Full Papers*				
NegBERT$_R$	*92.26*	**91.60**	86.00	81.25	78.57
NegBERT$_{XLM\text{-}R}$	91.64	89.68	84.19	81.89	79.40
NegBERT$_R$ (gold cues)	-	-	*92.91*	87.51	85.26
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	92.65	*88.99*	*87.81*
	Train: *SFU Review*				
NegBERT$_R$	80.65	78.98	69.35	65.29	62.81
NegBERT$_{XLM\text{-}R}$	*82.18*	*80.60*	69.19	66.54	64.15
NegBERT$_R$ (gold cues)	-	-	*85.64*	*82.08*	*82.14*
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	83.43	79.38	79.33

Table 11: Experiment results for NegBERT (F1-scores). Test set: *BioScope Abstracts*.

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: *BioScope Full Papers*				
	Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	*61.31*	*58.56*	44.78	22.03	20.24
NegBERT$_{XLM\text{-}R}$	58.30	55.70	41.03	18.80	18.10
NegBERT$_R$ (gold cues)	-	-	*74.20*	*41.58*	*37.54*
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	71.62	37.64	37.05
	Train: *BioScope Abstracts*				
NegBERT$_R$	*80.15*	*79.61*	69.26	65.02	64.01
NegBERT$_{XLM\text{-}R}$	79.91	78.97	66.72	64.03	61.29
NegBERT$_R$ (gold cues)	-	-	**92.06**	*83.71*	*83.36*
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	87.36	81.14	80.77
	Train: *BioScope Full Papers*				
NegBERT$_R$	**83.25**	**82.00**	71.60	64.43	64.43
NegBERT$_{XLM\text{-}R}$	83.14	81.81	69.12	71.47	70.44
NegBERT$_R$ (gold cues)	-	-	87.94	82.33	82.33
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	*89.41*	**89.17**	**88.85**
	Train: *SFU Review*				
NegBERT$_R$	*79.71*	*76.53*	66.26	59.63	59.45
NegBERT$_{XLM\text{-}R}$	78.39	75.24	66.09	59.81	59.51
NegBERT$_R$ (gold cues)	-	-	87.09	77.01	77.01
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	*87.28*	*80.44*	*80.44*

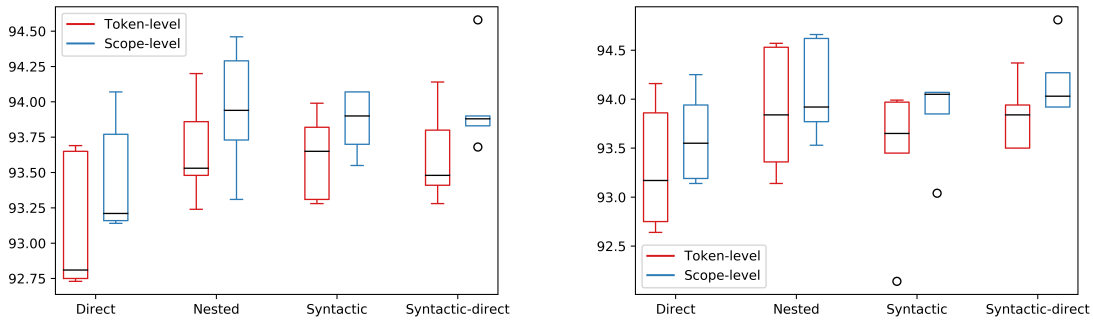
Table 12: Experiment results for NegBERT (F1-scores). Test set: *BioScope Full Papers*.

	Cue		**Scope**		**Full**
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
			Test: *SFU Review*		
	Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	62.35	53.48	54.13	11.62	11.36
NegBERT$_{XLM\text{-}R}$	*66.34*	*57.87*	56.01	14.51	14.79
NegBERT$_R$ (gold cues)	-	-	*78.06*	*33.09*	32.24
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	76.95	32.47	*32.99*
	Train: *BioScope Abstracts*				
NegBERT$_R$	*67.26*	58.25	65.75	62.98	47.46
NegBERT$_{XLM\text{-}R}$	60.01	*65.35*	60.82	57.47	53.45
NegBERT$_R$ (gold cues)	-	-	*86.05*	*84.78*	*84.78*
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	85.27	83.62	83.72
	Train: *BioScope Full Papers*				
NegBERT$_R$	*67.77*	58.32	65.87	62.46	46.47
NegBERT$_{XLM\text{-}R}$	61.75	*59.40*	61.65	56.72	46.17
NegBERT$_R$ (gold cues)	-	-	*84.36*	*82.30*	*80.73*
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	82.56	79.48	78.69
	Train: *SFU Review*				
NegBERT$_R$	81.60	80.92	73.63	74.34	72.51
NegBERT$_{XLM\text{-}R}$	**82.47**	**81.85**	74.50	74.91	73.60
NegBERT$_R$ (gold cues)	-	-	**88.87**	90.75	90.77
NegBERT$_{XLM\text{-}R}$ (gold cues)	-	-	88.69	**90.84**	**91.04**

Table 13: Experiment results for NegBERT (F1-scores). Test set: *SFU Review*.

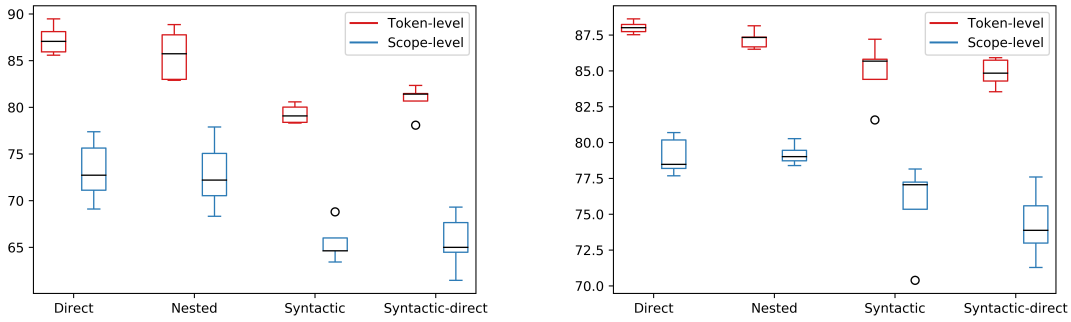
D *ConanDoyle-neg* F1-Score Distribution

The following figures depict the distribution of scores for various metrics for the STEPS models trained on *ConenDoyle-neg* (reannotated). Figure 19 depicts the F1-score distribution for the cue metrics, Figure 20 for the scope metrics, Figure 21 for the event metrics and Figure 22 for the full negation metric.



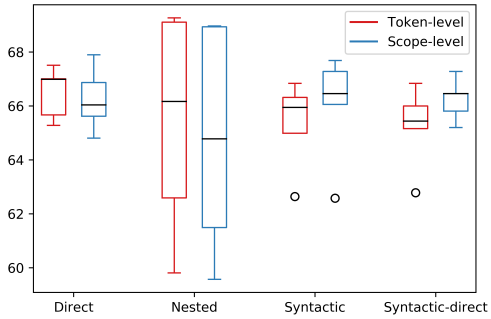
(a) Base settings. (b) Syntactic XLM-R.

Figure 19: The distribution of F1-scores for cue detection.

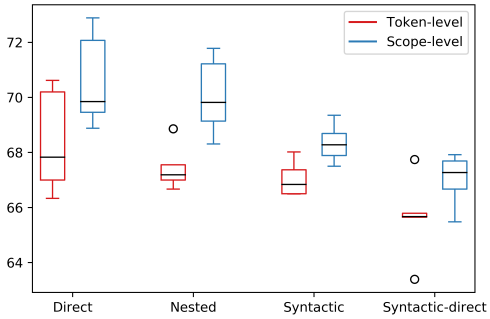


(a) Base settings. (b) Syntactic XLM-R.

Figure 20: The distribution of F1-scores for scope detection.

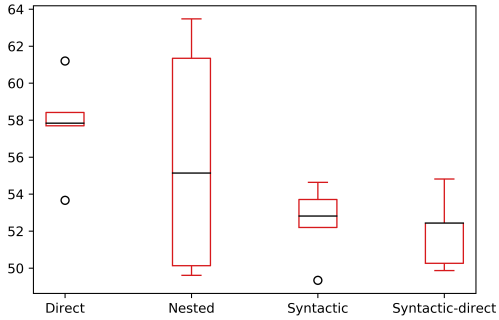


(a) Base settings.

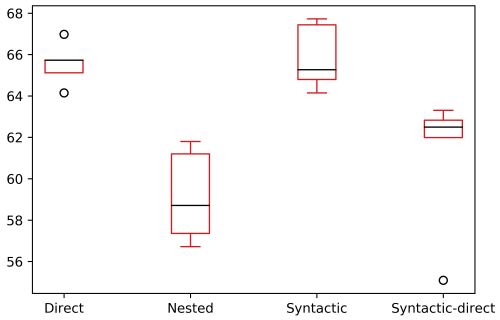


(b) Syntactic XLM-R.

Figure 21: The distribution of F1-scores for event detection.



(a) Base settings.



(b) Syntactic XLM-R.

Figure 22: The distribution of F1-scores for full negation detection.

E Cross-domain Results

The following tables report cross-domain results for the STEPS models (Grünewald et al., 2021) in comparison to the NegBERT$_R$ and NegBERT$_{XLM\text{-}R}$ models trained on various datasets and tested on a specific dataset. Table 14 reports results for testing on *ConanDoyle-neg* (reannotated), Table 15 for testing on *BioScope Abstracts*, Table 16 for testing on *BioScope Full Papers* and Table 17 for testing on *SFU Review*. The highest results are written in bold, the highest results among different mappings within the same settings are written in italics.

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: *ConanDoyle-neg* (reannotated)				
	Train: *BioScope Abstracts*				
NegBERT$_R$	**77.10**	**74.52**	**65.82**	**13.85**	11.02
NegBERT$_{XLM\text{-}R}$	71.40	67.72	61.70	13.03	**13.38**
Direct mapping	72.39	68.43	56.07	13.12	*12.96*
Nested mapping	71.09	66.47	53.91	13.64	10.74
Syntactic XLM-R					
Direct mapping	*73.02*	*68.88*	58.30	*13.76*	10.21
Nested mapping	72.52	68.11	*59.97*	13.71	10.54
	Train: *BioScope Full Papers*				
NegBERT$_R$	**76.09**	**74.66**	**64.39**	**14.41**	11.55
NegBERT$_{XLM\text{-}R}$	68.91	68.90	58.86	12.97	8.77
Direct mapping	68.87	67.95	49.17	*13.02*	**11.77**
Nested mapping	67.91	66.86	48.49	11.45	11.22
Syntactic XLM-R					
Direct mapping	70.34	69.53	*52.06*	10.39	10.55
Nested mapping	*71.50*	*71.06*	50.66	10.75	10.14
	Train: *SFU Review*				
NegBERT$_R$	77.07	**74.73**	**69.23**	14.55	14.97
NegBERT$_{XLM\text{-}R}$	**77.30**	74.38	68.24	**15.25**	14.67
Direct mapping	*76.36*	*73.84*	65.27	14.19	**15.23**
Nested mapping	74.98	72.43	65.05	13.97	13.34
Syntactic XLM-R					
Direct mapping	75.55	72.45	*67.20*	*15.04*	13.18
Nested mapping	76.11	73.16	66.43	14.98	14.41

Table 14: Experiment results (F1-scores). Test set: *ConanDoyle-neg* (reannotated).

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: *BioScope Abstracts*				
	Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	60.66	58.87	51.88	24.32	20.24
NegBERT$_{XLM\text{-}R}$	63.40	61.66	52.93	24.53	**21.84**
Direct mapping	67.19	65.57	54.53	*17.53*	4.82
Nested mapping	*69.38*	*67.79*	*54.62*	16.99	5.30
Syntactic mapping	68.89	67.29	46.76	13.34	*5.64*
Syntactic-direct mapping	69.02	67.41	45.39	9.84	2.05
Syntactic XLM-R					
Direct mapping	67.60	65.93	57.17	22.14	4.50
Nested mapping	68.12	66.48	**58.34**	**25.22**	5.22
Syntactic mapping	**69.80**	**68.16**	57.03	23.63	*8.75*
Syntactic-direct mapping	69.23	67.58	55.35	19.75	5.84
	Train: *SFU Review*				
NegBERT$_R$	80.65	78.98	69.35	65.29	62.81
NegBERT$_{XLM\text{-}R}$	82.18	80.60	69.19	66.54	64.15
Direct mapping	84.98	83.63	70.20	62.09	58.67
Nested mapping	**85.43**	**84.09**	71.70	63.24	60.16
Syntactic XLM-R					
Direct mapping	84.99	83.63	**72.78**	68.78	66.00
Nested mapping	85.39	84.05	72.74	**68.91**	**66.38**

Table 15: Experiment results (F1-scores). Test set: *BioScope Abstracts*.

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: *BioScope Full Papers*				
	Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	**61.31**	**58.56**	44.78	**22.03**	**20.24**
NegBERT$_{XLM\text{-}R}$	58.30	55.70	41.03	18.80	18.10
Direct mapping	57.27	54.82	41.68	14.54	5.13
Nested mapping	58.94	56.42	*43.71*	*15.47*	*6.64*
Syntactic mapping	58.90	56.44	39.28	14.81	3.77
Syntactic-direct mapping	*59.17*	*56.66*	39.36	12.52	2.93
Syntactic XLM-R					
Direct mapping	56.75	54.17	41.95	10.63	0.00
Nested mapping	57.19	54.59	46.67	12.25	0.53
Syntactic mapping	58.53	56.00	**47.01**	*21.93*	*9.07*
Syntactic-direct mapping	*58.75*	*56.16*	46.71	18.78	4.47
	Train: *SFU Review*				
NegBERT$_R$	79.71	76.53	66.26	59.63	59.45
NegBERT$_{XLM\text{-}R}$	78.39	75.24	66.09	59.81	**59.51**
Direct mapping	**79.79**	76.64	62.26	50.88	49.80
Nested mapping	**79.79**	76.64	65.24	54.17	52.15
Syntactic XLM-R					
Direct mapping	78.86	**76.72**	68.30	59.63	58.35
Nested mapping	**79.79**	76.64	**68.49**	**60.18**	*59.20*

Table 16: Experiment results (F1-scores). Test set: *BioScope Full Papers*.

	Cue		Scope		Full
Level	*Token*	*Scope*	*Token*	*Scope*	*Scope*
	Test: *SFU Review*				
	Train: *ConanDoyle-neg* (reannotated)				
NegBERT$_R$	62.35	53.48	54.13	11.62	11.36
NegBERT$_{XLM\text{-}R}$	66.34	57.87	**56.01**	**14.51**	**14.79**
Direct mapping	66.37	57.39	53.50	10.22	5.90
Nested mapping	66.64	57.55	*54.65*	*11.63*	*7.14*
Syntactic mapping	66.65	57.50	49.70	8.07	5.98
Syntactic-direct mapping	*66.78*	*57.65*	49.61	8.51	5.17
Syntactic XLM-R					
Direct mapping	66.41	57.44	*55.88*	12.35	5.61
Nested mapping	66.05	57.11	55.34	*13.22*	6.02
Syntactic mapping	66.70	57.64	53.21	12.59	7.39
Syntactic-direct mapping	**66.91**	**57.93**	53.37	12.36	*7.67*
	Train: *BioScope Abstracts*				
NegBERT$_R$	67.26	58.25	65.75	**62.98**	47.46
NegBERT$_{XLM\text{-}R}$	60.01	65.35	60.82	57.47	**53.45**
Direct mapping	61.24	**68.31**	59.41	54.00	*51.06*
Nested mapping	59.64	66.69	57.25	53.25	50.83
Syntactic XLM-R					
Direct mapping	**69.64**	63.39	**69.37**	*61.59*	48.07
Nested mapping	67.71	63.37	67.57	60.28	48.42
	Train: *BioScope Full Papers*				
NegBERT$_R$	67.77	58.32	**65.87**	**62.46**	**46.47**
NegBERT$_{XLM\text{-}R}$	61.75	59.40	61.65	56.72	46.17
Direct mapping	58.25	**63.53**	54.88	41.42	38.08
Nested mapping	57.24	62.87	55.12	42.33	*38.80*
Syntactic XLM-R					
Direct mapping	68.16	59.33	*60.89*	*48.93*	35.30
Nested mapping	**68.37**	59.62	59.50	43.66	32.10

Table 17: Cross-domain experiment results (F1-scores). Test set: *SFU Review*.