Natural Language Engineering **1** (1): 1–29. Printed in the United Kingdom

Comparison of Rule-Based and Neural Network Models for Negation Detection in Radiology Reports

D. SYKES,
Division of Psychiatry, Centre for Clinical Brain Sciences

A. GRIVAS, C. GROVER, R. TOBIN,
Institute for Language, Cognition and Computation, School of Informatics

C. SUDLOW
Usher Institute of Population Health Sciences and Informatics

W. WHITELEY
Centre for Clinical Brain Sciences, Edinburgh Medical School

A. MCINTOSH, H. WHALLEY
Division of Psychiatry, Centre for Clinical Brain Sciences

and B. ALEX
Institute for Language, Cognition and Computation, School of Informatics and
Edinburgh Futures Institute, School of Literatures, Languages and Cultures.

(*Received 22 May 2019; revised XXX*)

Abstract

Using natural language processing it is possible to extract structured information from raw text in the Electronic Health Record (EHR) at reasonably high accuracy. However, the accurate distinction between negated and non-negated mentions of clinical terms remains a challenge. EHR text includes cases where diseases are stated not to be present or only hypothesised, meaning a disease can be mentioned in a report when it is not being reported as present. This makes tasks such as document classification and summarisation more difficult.

We have developed the rule-based EdIE-R-Neg, part of an existing text mining pipeline called EdIE-R (Edinburgh Information Extraction for Radiology reports), developed to process brain imaging reports,[1] and two machine learning approaches; one using a bidirectional long short-term memory network and another using a feedforward neural network. These were developed on data from the Edinburgh Stroke Study, and tested on data from routine reports from NHS Tayside (Tayside). Both datasets consist of written reports from medical scans.

These models are compared with two existing rule-based models; pyConText [Harkema et al., 2009], a python implementation of a generalisation of NegEx, and NegBio [Peng

[1] `https://www.ltg.ed.ac.uk/software/edie-r/`

et al., 2017], which identifies negation scopes through patterns applied to a syntactic representation of the sentence. On both the test set of the dataset from which our models were developed, as well as the largely similar Tayside test set, the neural network models and our custom-built rule-based system outperformed the existing methods.

EdIE-R-Neg scored highest on F1 score, particularly on the test set of the Tayside dataset, from which no development data was used in these experiments, showing the power of custom-built rule-based systems for negation detection on datasets of this size.

The performance gap of the machine learning models to EdIE-R-Neg on the Tayside test set was reduced through adding development Tayside data into the ESS training set, demonstrating the adaptability of the neural network models.

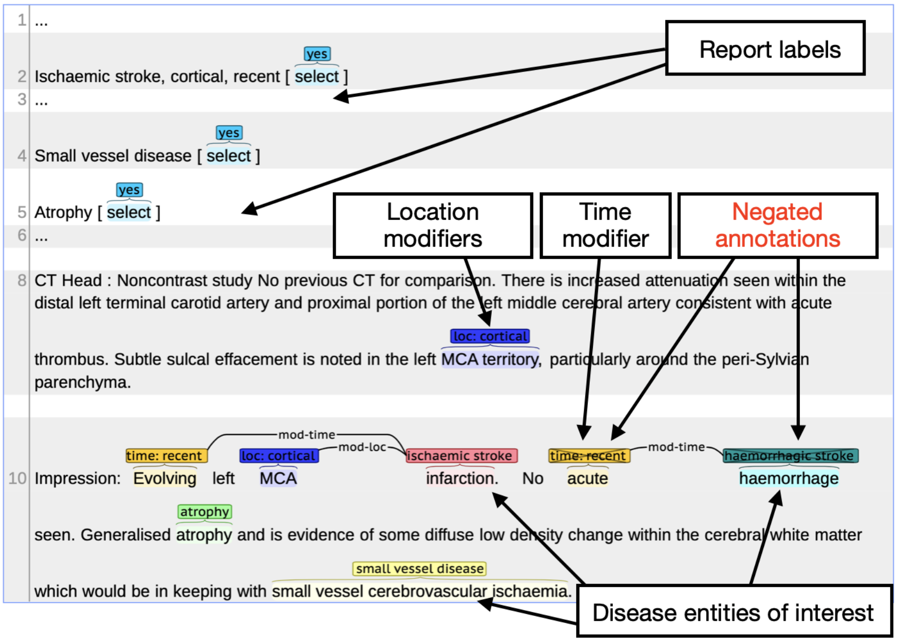
1 Introduction

The goal of natural language processing (NLP) is to analyse and understand text data automatically. Negation detection is a sub-problem within NLP that consists of identifying negation cues and their scopes. For our objectives, we treat negation detection as assertion of whether the entity in question is present or absent, where ambiguous cases are treated as absent. Entities are key terms, such as disease names, and location or time modifiers of those diseases. For example, in the excerpt "*there was no tumour present. The same is true for atrophy but there is evidence of an acute ischaemic stroke*", both *tumour* and *atrophy* are negated and *ischaemic stroke* is not. *Acute* is a time modifier and is also not negated as it refers to the non-negated *ischaemic stroke.*

We apply negation detection to the analysis of radiology reports, written natural language text describing findings and observations of medical health professionals. We used the Edinburgh Stroke Study (ESS) and the National Health Service Tayside (Tayside) datasets, which both consist of reports of radiology scans of the brain. Our entities on which we do negation detection are names of diseases, the anatomical location of the disease, and whether development of the disease is recent or old. Due to the sensitivity of the data, we provide a realistic synthetic example of a radiology report in Figure 1, where negated entity annotations are crossed out.

The work presented here is part of an MRC Mental Health Data Pathfinder project on linking data extracted from brain imaging reports to EHR data on mental health with the aim to study correlations between physical and mental health in the Generation Scotland cohort.[2] One aim of the NLP work in this Pathfinder project is to label radiology reports with an indication of what was observed by the radiologist, such as disease type, location of the disease and whether the disease is recent or old. This is a text mining task involving named entity recognition, negation detection, relation extraction and document classification. We are developing a rule-based system called EdIE-R (Edinburgh Information Extraction for Radiology Reports) containing these four processing components [Alex et al., 2019]. We are also experimenting with machine learning alternatives for each component making

[2] `https://www.ed.ac.uk/generation-scotland`



...

Ischaemic stroke, cortical, recent [select]

...

Small vessel disease [select]

Atrophy [select]

...

CT Head : Noncontrast study No previous CT for comparison. There is increased attenuation seen within the distal left terminal carotid artery and proximal portion of the left middle cerebral artery consistent with acute thrombus. Subtle sulcal effacement is noted in the left MCA territory, particularly around the peri-Sylvian parenchyma.

Impression: Evolving left MCA infarction. No acute haemorrhage seen. Generalised atrophy and is evidence of some diffuse low density change within the cerebral white matter which would be in keeping with small vessel cerebrovascular ischaemia.

Fig. 1. An example of a synthetic yet realistic brain imaging report with disease entity, modifier, negation, relation and label annotations created for the ESS and Tayside datasets. Manual annotation used for development, training and evaluation were conducted using the Brat annotation tool [Stenetorp et al., 2012].

use of neural networks. We have previously presented a method comparison for the named entity recognition step [Gorinski et al., 2019], and showed that it is difficult to outperform a rule-based system specifically designed for recognising named entities in brain imaging reports.

In this paper we focus on the negation detection step. We assume that the disease entity and modifier spans are known and compare each model's ability to predict on the unseen test sets whether these disease entity and modifiers are negated or not. We also provide an analysis of negation in our data. We compare three rule-based methods and two based on neural networks:

- pyConText, a rule-based negation detection tool which is available as a Python implementation of ConText, a variation of the widely used NegEx algorithm (see Section 5.1),
- NegBio, a rule-based tool for negation and uncertainty detection in clinical texts (see Section 5.1),
- EdIE-R-Neg, a rule-based negation detection component in EdIE-R which was specifically developed for analysing brain imaging reports (see Section 5.2),
- FFNN-Neg, a feedforward neural network trained on brain imaging reports for negation detection (see Section 5.4), and

- BiLSTM-Neg, a bidirectional recurrent neural network trained on brain imaging reports for negation detection (see Section 5.5).

For each approach, we report overall negation detection results and provide analysis where performance varied by entity type and sub-type.

Our models are developed, trained and tested on two datasets of radiology reports from the Edinburgh Stroke Study (ESS; n=630) [Jackson et al., 2008] and NHS Tayside (Tayside, n=1,062). These were manually annotated for 12 named entity types and 4 modifiers. Training and development was largely done on the development subset of the ESS dataset, and evaluation was carried out on unseen test subsets of both.

To test generalisation and the effectiveness of porting our models to similar datasets, we ran cross-training experiments on the machine learning models, evaluating whether introduction of parts of the Tayside development data into the training set (transfer learning [Pratt et al., 1991]), increased performance on the Tayside test set.

The experiments described in this paper are preceded by a corpus analysis to illustrate the nature of negation cues in such data. From the corpus analysis we discover that the negation detection task as we frame is relatively straightforward, due to factors such as not needing to identify scope, and treating uncertain cases as negative. In addition, negation patterns in radiology reports are known to be more simple than in other documents. We treat negation as a binary classification or assertion of definite presence of medical entities (presence vs. non-presence), as is often done in clinical NLP. This application is different to how negation is generally conceptualised in linguistics literature. It was chosen due to the wider project aims of document classification and labelling radiology reports with an indication of what was observed by the radiologist, for which binary classification is more useful than other approaches that focus on identifying scope or ambiguous cases. That this conceptualisation makes the task less complex is evidenced by the high F1 scores of our models.

2 Motivation for using Neural Networks for Negation Detection

Rule-based NLP models are often employed for information extraction and labelling of raw text documents, particularly in the medical field where labelled training data is often sparse or non-existent (e.g. see examples in Pons et al. [2016]). As we initially had no annotated data available for the purpose of text mining brain imaging reports, the rule-based EdIE-R system was developed as a starting point. In parallel with writing rules for EdIE-R, data from the Edinburgh Stroke Study and NHS Tayside was annotated manually by domain experts to create gold standard data for evaluation.

The annotated datasets, which we now have available and are used for the experiments presented here, amount to over 1,600 reports containing over 13,000 entity annotations in total. This means that the use of machine learning, and in particular neural networks, becomes more viable. Rule-based systems, while highly effective, are time consuming to set up and can sometimes translate poorly between datasets. In this paper we show that an effective neural network can give near equivalent performance when used for negation detection. Once an effective neural network model has been found we would expect it to have the benefit of being quicker to adapt to new datasets that differ more greatly than the two analysed here, as hyperparameters and architecture should translate reasonably between datasets meaning only fine-tuning is necessary.

In addition to the rule-based model, we have therefore developed neural network models which learn to detect negation by capturing negation patterns in the annotated data. In both cases we treat negation detection as binary classification limiting our predictions to entity tokens, the spans of which we assume are already known. Here we test two neural network architectures, a feedforward neural network (FFNN) and a bidirectional long short-term memory (BiLSTM) neural network.

Similar to previous work [Fancellu et al., 2016], we begin our investigation using a feedforward network as a baseline neural network model. Apart from being a simpler and faster model to train than BiLSTMs, using a feedforward network is appealing as controlling the amount of surrounding context used to predict negation is achieved easily. While BiLSTMs are more powerful models that can learn what salient information to store in their internal state, we assume they may need more data to do so. Therefore, truncating the input for use with a feedforward network may be beneficial in a low resource setting. In their related work on negation scope detection, Fancellu et al. [2016] used window sizes of 9 and 15 tokens, as for their English literature datasets, it was found that 95% of words in the negation scope occurred within a window of 9 tokens to the left and 15 to the right of the negation cue. As we shall see in Section 4.1, in the ESS radiology report dataset the context window containing the negation cues most useful for negation detection is skewed to the left, but also has tails that taper off at similar distances from the negated token, 15 tokens to the left and 10 to the right.

LSTMs [Hochreiter and Schmidhuber, 1997], the second type of neural network approach evaluated here, are recurrent neural networks (RNNs). RNNs are often employed in NLP tasks due to their effectiveness at modelling sequences of arbitrary length. This makes them in theory able to take advantage of long distance dependencies in the input, which would be impossible to capture with any finite window feedforward or convolutional neural network. However, traditional RNNs were demonstrated to suffer from exploding/vanishing gradients [Bengio et al., 1994], an issue that impedes learning and limits the effective maximum distance of dependencies that can be learned in practice.

LSTMs are an augmented RNN architecture that ameliorates the exploding/vanishing gradients issue through use of an internal gating system. The gating system controls the flow of information to and from the hidden state/memory of the network, demonstrably improving its ability to capture distant dependencies. BiLSTMs [Graves and Schmidhuber, 2005] are a further extension to the LSTM model that take into account both the left and the right context of an input when constructing its representation. This property is very important for negation detection, as negation cues can precede or succeed a target word in our dataset, as we will show when analysing negation cue patterns in Section 4.1.

3 Related Work

A popular approach for Negation Detection is the use of regular expressions, such as NegEx [Chapman et al., 2001] which is often considered as a benchmark. This type of approach works by matching patterns that indicate negation. To increase performance NegEx also filters out sentences that appear to be falsely negated and sets limits to the scope of negation phrases.

While the original implementation of NegEx limited scope by number of words, an improved version called ConText increased scope to the entire sentence and in addition to predicting negation scopes, can also predict other contextual properties of clinical conditions such as whether the disease is historical or hypothetical [Harkema et al., 2009].

Variations of NegEx have been applied by multiple teams to the medical domain [Harkema et al., 2009; Cornegruta et al., 2016; Horng et al., 2017]. Subsequently, we use a Python implementation (pyConText) of the ConText algorithm as the first baseline for our experiments.

Another early method involving the development of a set of rules for recognising negation patterns in text involved a lexical scanner followed by a parser that uses a restricted subset of context-free grammars called LALR(1) grammars [Mutalik et al., 2001]. This and the regular expression model are both relatively simple approaches, but nonetheless capture a large number of negations in the text and work effectively when compared with classification based approaches [Goryachev et al., 2006].

Regular expression based methods such as those used in NegEx have been expanded upon more by introducing a grammatical parser [Huang and Lowe, 2007]. This method successfully addresses the issue of negation cue phrases being more than a few words from the entity they negate, where they would fall out of the scope of methods that only use regular expressions. Another method, DEEPEN [Mehrabi et al., 2015], uses a dependency parser to also improve upon results from NegEx.

NegBio [Peng et al., 2017] is a more recent method developed for radiology reports that makes use of a dependency parser, removing the scope limitation of the simpler methods through utilisation of universal dependencies for pattern definition and sub-graph matching for graph traversal search. This use of a dependency parser resulted in an improvement over NegEx of 5.1% in F1 Score, and it has also been used in development of a chest X-ray database [Wang et al., 2017]. We use NegBio as our second baseline algorithm.

Previous attempts to use a machine learning approach in place of rule-based systems have focused around more traditional machine learning methods, including Support Vector Machines (SVM) [Cruz et al., 2017]. A system trained on an opinion mining corpus used SVMs for negation cue and negation scope detection as two distinct sub-problems, as was common with the rule-based methods [Cruz et al., 2017].

We present experiments using two neural network approaches for negation detection, one of them using a bidirectional LSTM architecture. A similar method has been used for negation detection in Electroencephalography Reports [Taylor and Harabagiu, 2018]. Another approach that made use of LSTMs as well as a deep rectified linear network also used multi-task learning for tasks including negation detection in EEG reports, achieving promising results [Maldonado et al., 2017].

In the radiology report domain, Cornegruta et al. [2016] also used a bidirectional LSTM approach for mention extraction and negation detection in a corpus of chest X-ray reports. Their bidirectional LSTM model significantly outperformed a rule-based system that used ontology dictionary lookups (RadLex, MeSH) in addition to fuzzy matching through string similarity measures. However, for negation detection a NegEx variant that leveraged Stanford CoreNLP [Manning et al., 2014] to strip sentences of all words not held together by negation and conjunction dependency arcs outperformed the syntax-unaware implementation of NegEx, as well as the BiLSTM model which also does not make use of syntax.

Most recently, Peng et al. [2019] reported on a method to predict relevant, irrelevant and uncertain mentions of lesions in radiology reports. While this task is different to negation detection, there is some overlap since uncertain mentions may be expressed using negation cues. They train a self-attention convolutional network on sentences and post-process its output using a rule-based system. They report that their rule-based system has very high precision but low recall, and they therefore demonstrate that using it as an additional filter on the output is beneficial.

The main contribution of the work presented here is a comparison of three rule-based approaches to two neural network based approaches for negation detection in brain imaging reports. BiLSTMs have been applied and evaluated previously for this task on radiology reports of chest X-rays but not for brain images, which differ in aspects such as annotation guidelines, named entity characteristics, and lexical and syntactic context, which can cause changes in performance [Wu et al., 2014]. Our experiments were conducted using significantly less training data than previous work but the results are nevertheless promising. We also present a corpus analysis of negation cues in our data which guided model development.

4 Datasets

In this paper, we present experiments on negation detection using different approaches developed, trained and evaluated on Scottish radiology reports from two sources (Edinburgh Stroke Study (ESS) and Tayside). While the ESS data consists of anonymised radiology reports from brain CT and MRI scans conducted as part of the Edinburgh Stroke Study (n=1,168), the Tayside data is a subset of reports for routine CT and MRI scans created in National Health Service (NHS) Tayside (n=156,619).

From each collection, a subset of reports (ESS: n=630, Tayside: n=1,062) was manually annotated by domain experts (a neuroradiologist and a neurologist) using the brat annotation tool [Stenetorp et al., 2012]. The manual annotation consists of the labels, entities, modifiers, negation and relation mark-up illustrated in Figure 1.

Tayside data was received in parts, with development data being taken from the first part received and selected to match keywords for our target entities such as *bleed* and *haemorrhage*, as these are low frequency in routine reports. The Tayside test set was taken from the fourth part, and was selected at random to produce a dataset with a distribution of entities that is more representative of the data we would expect to see in future. Though the parts were expected to be consistent in entity distribution, they varied, and this in addition to the selection criteria led to differences in disease entity distributions between the development and test data, as seen in Table 2.

The difference in the data (hospital-based register in the case of ESS versus routine scans in the case of Tayside) becomes apparent in the number of tokens, sentences and entities in each dataset. ESS contains more annotated entities than Tayside, while Tayside contains a lot more tokens, but not sentences, than ESS. This suggests that the radiologists writing the Tayside reports used longer sentences. Because the Tayside data is not specific to stroke and tumour patients, it is not surprising that the number of entities annotated in them is lower.

As can be seen, entities include two types: disease observations and modifiers. The full set of disease observations are: atrophy, haemorrhagic stroke, haemorrhagic transformation, ischaemic stroke, meningioma (mening tumour), metastatic tumour (metast tumour), microhaemorrhage, small vessel disease, stroke, subarachnoid haemorrhage, subdural haematoma and tumour. Modifiers designate a time or anatomical location for the disease and are further split into these two sub-types. Location can be cortical or deep, and time can be recent or old. Word tokens in a report can be associated with either a disease observation, a modifier or both, and all modifiers are associated with a disease via a relation. All entities appear with either a negated or non-negated annotation in the data. We note that we only annotate negation on entity and modifier annotations and therefore a token that is not an entity or modifier cannot be marked as negated.

There are many cases where the language expresses uncertainty rather than a clear positive/negative decision. For example, "*it is not possible to exclude a small acute focal infarct*", while containing an overt negation cue word, is non-committal as to the presence of an infarct. Furthermore, sentences with no overt negation cue may also be non-committal, e.g. "*a small acute or subacute infarct may be missed*". In these cases the annotators were instructed to mark the entities as negated since the clinician has not clearly affirmed their presence, treating negation as assertion of whether the entity is definitely present or not. This treatment of negation is chosen due to our overall project objectives.

The annotation scheme was developed iteratively over a first tranch of the ESS development data with the annotators correcting the output of an early version of the EdIE-R rule-based system. This tranch of data (123 reports) was doubly annotated, inter-annotator agreement (IAA) was monitored and the annotators discussed and reconciled differences to create a jointly-agreed initial dataset. This dataset was used to refine the rules in EdIE-R, which in turn fed into subsequent rounds of annotation. The remainder of the ESS development set was singly annotated, but the entire ESS test set (266 reports) was doubly annotated and IAA measured using precision, recall and F1 score. All of the Tayside data was subsequently singly annotated apart from the last 100 reports in the test set, which were doubly annotated to monitor IAA.

Since we only annotated negation for annotated entities, we cannot report IAA for negation and entities independently. We therefore report IAA for entities and negation combined. Since we are effectively computing IAA for named entity recognition with twice as many entity types, we report F1-score [Hripcsak and Rothschild, 2005]. The combined NER and negation IAA F1 score on the ESS test set was 96.52 and 95.52 for Tayside. In order to get an additional estimate of IAA for negation detection, we can isolate negation scores for all entities the annotators agreed on. In this case we report Cohen's kappa [Cohen, 1960] since we are effectively reporting agreement on binary classification, albeit on a subset of all entities. The negation IAA was $\kappa = 99.18$ for ESS and $\kappa = 100.00$ for Tayside, which equate to almost perfect and perfect agreement respectively.

Table 1 lists the numbers of reports, sentences and words for each dataset, as well as total number of entities, disease entities and modifiers. The annotated reports in the ESS dataset were split into a development set (364 reports and 4,332 entities), and an unseen test set (266 reports and 2,924 entities). The Tayside data was split in a similar way (362 reports for development and 700 reports for evaluation). Rules were designed using the development portion of the ESS dataset.

For our neural network approaches we use 80% of the ESS development documents for training our models and the remaining 20% of the ESS development documents as a validation set to choose the best hyperparameters for our models. Transfer learning where model weights trained on one dataset are used for another is an increasing focus in NLP [Mou et al., 2016], as well as machine learning as a whole, and we also ran transfer learning based experiments incorporating portions of the Tayside development documents into our training set for the neural networks, to evaluate adaptation to a novel dataset from that used for hyperparameter selection and training. We chose to train primarily on the ESS development data to allow comparison with the rule-based model. We then reported the performance of each of our models on both unseen test sets. The subsets of each dataset (ESS and Tayside) are as follows:

- Development data, the part of the datasets that is used for development of the rule-based system, and for the training process (training and hyperparameter selection) of the machine learning models. The development data encompasses both the training and validation data used by the machine learning algorithms,
- Training data, the part of the datasets that is used to train the machine learning models,
- Validation data, the part of the datasets used for hyperparameter selection for the machine learning approaches, and
- Test data, the part of the datasets that remains completely unseen by the models until test time.

	ESS Dev	ESS Test	Tayside Dev	Tayside Test
Reports	364	266	362	700
Sentences	3,837	2,855	2,791	3,948
Tokens	32,229	22,842	50,522	48,519
Total Entities	4,332	2,924	2,997	2,986
Disease Entities	2,373	1,494	1,361	1,501
Modifier Entities	1,959	1,430	1,636	1,485

Table 1. Dataset statistics.

Dataset	Disease Entities POS / NEG	Modifier Entities POS / NEG	Total POS / NEG
ESS dev	1,320 / 1,053	1,621 / 338	2,941 / 1,391
ESS test	836 / 658	1,127 / 303	1,963 / 961
Tayside dev	1,002 / 359	1,549 / 87	2,551 / 446
Tayside test	654 / 847	1,268 / 217	1,922 / 1,064

Table 2. Disease and modifier entity negation statistics across our datasets. The leading count is the number of non-negated occurrences (POS) followed by the number of negated occurrences (NEG).

4.1 Negation Analysis

In this section we begin by investigating how negation is expressed in brain imaging reports, in particular in the ESS development set. As we shall demonstrate, the manifestations of negation in such reports is clearly marked, making negation detection in this setting much more straightforward than negation detection in general purpose language, where it can be quite challenging. Namely, in the ESS dataset, negation is predominantly introduced explicitly using *no* and the negation scope is extended through connectives such as *or*, *and* and *punctuation*, as in the example "*No acute haemorrhage, masses or extra-axial collections*". For brevity, we shall refer to all such tokens that are correlated with the introduction of negation as cues.

In Table 2, we tabulate the number of negated and non-negated disease entities and modifiers. For this study, negation cues were not explicitly annotated, so in the following analysis, we use the number of times a token appears in the context window of a negated disease or modifier annotation as a proxy for identifying negation cues.

In order to get a better understanding of the types of negation cues used, in Table 3 we report the counts of cues that occur in the context of negated disease and modifier annotations. We group negation cues into three groups depending on their function. The first group of cues introduces negation (Negation Cue), the

Negation Cue		Connective		(Un)certainty Cue	
token	counts	token	counts	token	counts
no	626	or	376	evidence	137
but	11	,	291	identified	30
any	8	and	23	demonstrated	20
against	7	/	13	definite	13
nil	6	with	11	suggest	7
cannot	4	-	7	may	5
however	3	nor	2	would	4
not	3	ro	1	likely	4
although	2			appears	2
				attributable	1
				might	1
				obvious	1

Table 3. Counts of most common tokens that surround a negated mention of a disease/modifier. Tokens are grouped into types depending on the manner with which we interpret them to interact with the mentions. They are ordered from top to bottom from most frequent to least frequent. The counts were constructed using a window of the 15 previous and 15 next tokens as context.

second extends negation to more entities through connectives (Connective) and the third contains cues that express certainty or uncertainty ((Un)certainty Cue). Cues with such characteristics seem to be a common finding when working with radiology reports, see for example Peng et al. [2019, Table 2] which targets regular expression patterns such as *no evidence of* and *or/and* among others.

The counts in Table 3 show that in the overwhelming majority of cases, negation is explicitly introduced using *no* with the remaining negation cues being rather scarce. From the middle column of our table, we extrapolate that negation scope is often extended to more entities through the use of connectives, such as *or*, *and*, *nor*, and the use of punctuation such as commas. In the last column, we tabulate more cues that potentially introduce certainty or uncertainty. For our study, we assume these cases also introduce negation, since annotators were instructed to mark all mentions of diseases and related modifiers that are not clearly indicated present as negated.

Lastly, we also inspected the coverage of these tokens, namely how many cases of negation in the dataset can be explained by finding a single occurrence of any of the cues listed in Table 3. We note that the presence of connectives doesn't necessarily imply negation, however, as we shall see in Figure 2, connectives such as *or* and *comma* occur much more frequently surrounding negated entities than positive entities. Hence connectives contain a lot of information relating to negation in our dataset. We found that when inspecting a symmetric context window of 15 previous tokens and 15 next tokens surrounding a negated entity or modifier, we could not find any cues in the context in only 3 out of 1387 cases. When we narrowed

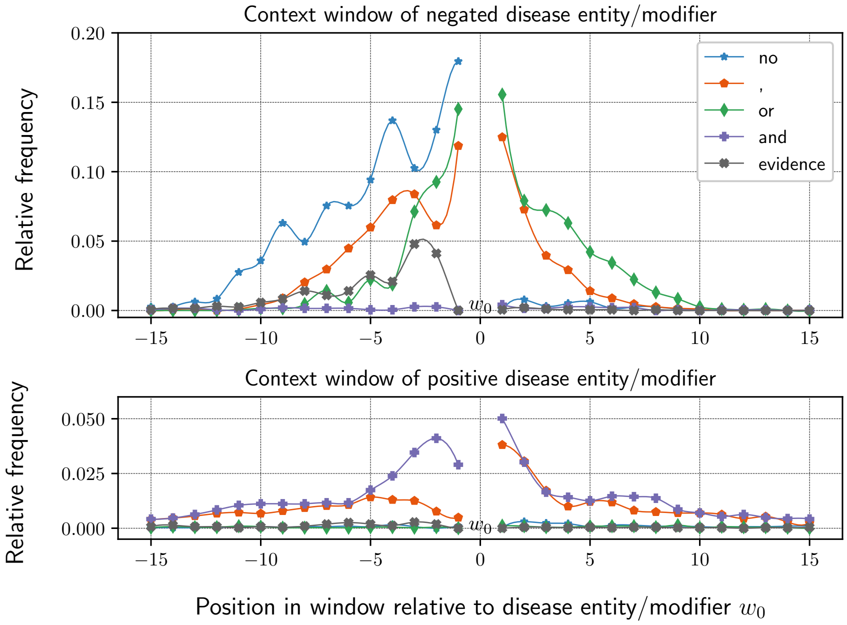


Fig. 2. Negation cue distribution around negated (top graph) and positive (bottom graph) disease entities and modifiers for a selection of most frequent cues from Table 3. We have plotted interpolating lines to highlight the relative increase and decrease of frequency depending on the position in the context window. We note that for the negated contexts seen in the top graph, *no* and *evidence* almost always precede the mention while other cues appear both before and after it. Moreover, the appearance of *no* in the negated context can precede the entity by more than 10 tokens while connectives such as *and*, *or* and *comma* appear in a closer context, suggesting that they are used to extend the scope of negation. As expected, the cues are more frequent in negated contexts than positive contexts as can be seen by comparing the graphs (note that the y-axis is on a different scale).

down the context window to a symmetric window of 5 tokens, we found 9 cases that couldn't be explained by the presence of a cue. This reinforces our belief that we are dealing with a rather degenerate case of negation compared to that of negation in a more general domain: we can get good results by relying on the presence of a rather small list of cues.

Our above interpretation of connectives extending negation scope, becomes more evident when taking into account the top graph in Figure 2. The frequencies plotted are relative, in the sense that they are obtained by dividing the counts of our selected negation cues (*no, comma, or, and, evidence*) at a specific position in the context window by the number of non disease entity/modifier tokens aggregated over the whole window. This calculation is performed twice, once for negated con-

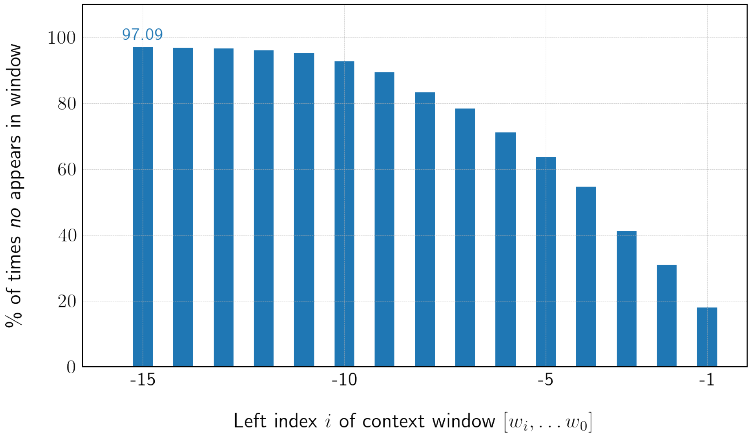


Fig. 3. Percentage of times negation of a disease entity or modifier t_0 can be explained by finding an occurrence of the negation cue *no* in the n preceding tokens. The graph can be interpreted as the cumulative value of the blue curve in the top graph of Figure 2, being aggregated from index 0 to -15.

text windows (top graph) and once for positive context windows (bottom graph). Therefore, the graph captures both the distribution of cues around a disease entity or modifier t_0 in the context window, as well as their relative frequency of occurrence. As can be seen, *no* can precede the negated entity by more than 10 tokens, while connective cues such as *or* and *comma* appear closer to the negated disease and modifier annotations. Moreover, *no* is by far the most common negation cue with connectives being the next most common. Lastly, we turn to the bottom graph that shows the relative frequencies of negation cues in positive contexts. As expected, negation cues in non-negated contexts are rare, with the exception of *and* and *comma* which can be used in positive contexts as well.

Given how commonly *no* occurs in the context of negated disease entities and modifiers, we now assess how often we can attribute negation to its appearance. As can be seen in Figure 3, *no* can be found in the 5 tokens preceding an entity around 50% of the time and in the preceding 15 tokens 97.09% of the time. This reinforces our understanding that negation is introduced explicitly and such negation cues are the most salient for negation detection.

The simplicity and direct way with which entities are negated in the ESS dataset comes as no surprise when one considers the purpose of clinical text. Namely, to communicate observations and diagnoses from a CT or MRI scan quickly and accurately to other clinicians. It is, however, not obvious that all teams of clinicians will communicate findings and express uncertainty in exactly the same way as that in the ESS dataset. For example, we saw earlier that *not* only occurs in the context of a negated word 3 times in the ESS dataset. Quite clearly, we could have a dataset where *not* is used to introduce negation commonly and in a variety of ways, a scenario which should give our current machine learning models difficulties, given they only have 3 examples to learn from. In the following section we shall introduce our models and investigate the following question: how well do machine learning models trained only on the ESS dataset transfer to the unseen Tayside dataset?

5 Negation Detection Algorithms

In this section, we describe each of the approaches (three rule-based algorithms versus two neural network algorithms) we used for negation detection in radiology reports in more detail.

5.1 Existing Rule-Based Methods: pyConText and NegBio

NegEx is an NLP algorithm designed for negation detection [Chapman et al., 2001]. Context [Harkema et al., 2009] is a generalisation of the NegEx algorithm, and here we adapted a Python implementation of ConText called pyConText for use with our dataset as a baseline. The algorithm uses a list of targets, for example disease names, and a list of modifiers that are applied to these targets, for example negation.

To use the pyConText algorithm with our dataset we created a list of targets based on the diseases in our dataset. To create the same output format as that of the other models compared in this paper, we made a separate target list of our anatomical location and temporal information entities. We used the modifier list distributed with the algorithm, however as we are only interested in negation of our entities we removed all modifiers from this list that were not involved in assigning negation.

These target lists use regular expressions (Regex) for identification of targets. To create the Regex for each target, the phrases identified by the expert in our reports are used in an *or* arrangement. For example for the target *microhaemorrhage* an expert has identified it appearing in the reports through the use of the phrases *micro bleeds* and *micro haemorrhage*, leading to the Regex **micro haemorrhage|micro bleeds**.

From the pyConText output negation information is applied on the word level, and this is combined with the entity data without negation information from the manually annotated data for the final output.

For our second baseline we used NegBio, and to use the NegBio [Peng et al., 2017] algorithm with our dataset we added targets from our dataset to the existing text target files, to ensure the named entity recognition part annotated all our entities. The annotations were then pruned so we were left with only the ones that matched with the manually annotated entities, and these were fed to the NegBio negation algorithm which adds negation information for each annotation.

For both of these methods we did not change any rules or logic for the negation detection part of the algorithms, focusing only on ensuring they effectively worked with our datasets and target entities. This was to allow comparison of our custom-built rule-based model with out-of-the-box models.

5.2 Rule-Based Model (EdIE-R)

The EdIE-R rule-based system has a pipeline architecture where mark-up is added by each component feeding on information provided by earlier processing. Early linguistic processing components include sectioning, tokenisation, sentence-splitting, part-of-speech tagging and lemmatisation. During these early stages, negation cue words such as *no*, *not*, *never*, *nor* are marked as negated. At this point the main information extraction components are applied, i.e. named entity recognition (NER) to mark up the entities, and relation extraction, which links disease entities and modifiers. In between these two steps, a shallow syntactic analysis known as chunking [Grover and Tobin, 2006] is performed followed by rules to determine the scope of negation.

In a simple case such as "*No obvious mass lesion*", the chunker establishes that this is a noun group, and negation, encoded as an attribute in the XML data structure, is propagated from *No* to the entire noun group. As *mass lesion* has already been identified by NER as a disease entity, this entity is marked as negated. Other cases require the chunker to recognise an embedded structure for noun groups, for example where *of*-phrases act as noun modifiers ("*No evidence of metastatic disease*") or disjunctive structures ("*No acute haemorrhage, masses or extra-axial collections*") or a combination of the two ("*No evidence of acute infarct or bleed*"). In these cases the negation is propagated to the entire noun group, thereby defining the scope of the negation cue.

Elsewhere the negation originates in a verb group and scopes over related noun groups, e.g. "*This does not show significant mass effect*" or "*A small recent infarct cannot be detected*". In such cases the identity of the verb is significant: in the above examples, negated *show* and *detect* lead to entities in their scope being identified as negated, but in "*The cerebellar haematoma has not increased in size*", the *haematoma* entity should not be negated.

Our treatment of negation doesn't identify uncertainty, treating uncertain cases as negative, and the EdIE-R negation rules were formulated accordingly. In the remainder of this paper, we refer to this method as EdIE-R-Neg.

5.3 Neural Network Approaches

In this section we will outline our two neural network approaches. We create word embeddings to encode each sentence's lexical information. In order to encode entity information for the target entity, we suppress the lexical information contained in the word tokens that represent the entity and only include an entity surrogate embedding, *ENT*. Namely, we include a learned embedding that encodes whether the tokens are an entity. If an entity uses more than one token, for example "*ischaemic stroke*", both of these tokens are replaced with a single *ENT* token. We found using surrogates for entity phrases in this way improved our results on the validation set, likely as the ratio of negated to non-negated entities is highly variable when split by entity and doesn't generalise.

Additionally, we pad the input to a prespecified maximum input length using a

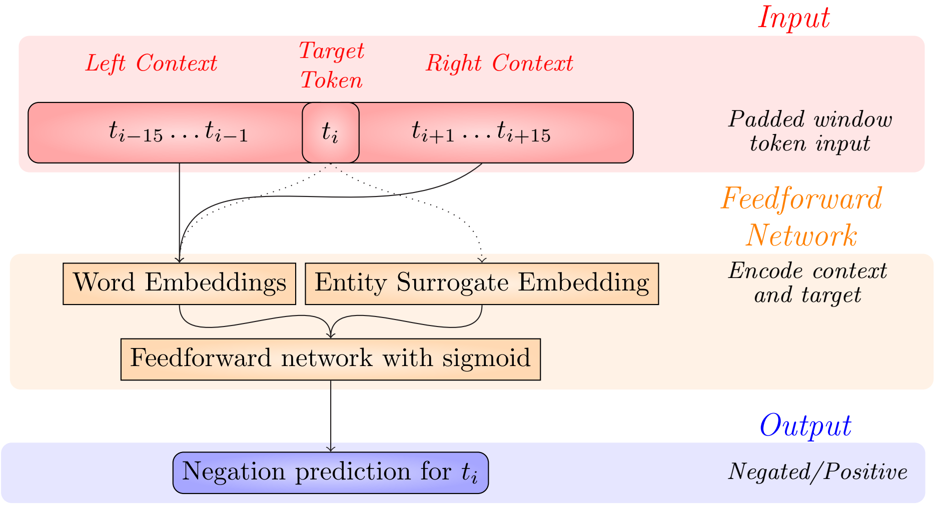


Fig. 4. FFNN-Neg architecture description. Given we wish to predict whether a target input t_i is negated, we feed a padded context window surrounding the target token to the network. To encode entity information, we replace the tokens of an entity with a single surrogate *ENT* token. We feed the concatenation of the embeddings of the context window through a feedforward network with a single hidden layer and a sigmoid output layer for negation prediction.

reserved *PAD* token that has its own embedding. Lastly we use the token *UNK* for any word not in our vocabulary. The vocabulary is built from the development data and we drop the least frequent 3% of the total words to train the *UNK* token embedding. We lowercase all words in our datasets.

These embeddings are fed through the networks as input. Both networks have an output layer with a sigmoid activation for binary classification. We use binary cross-entropy as our loss function. We train using the Adam [Kingma and Ba, 2015] optimizer with exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We randomly initialised word embeddings to be Gaussian $\sim \mathcal{N}(0, 1)$. We train until there is no improvement in F1 score on the validation data, a held out split of the development data.

The model is trained and evaluated using data manually annotated by human experts. The development data was split into 80% training and 20% validation. Models were run 5 times with random initialisations, with an average F1 score being used for hyperparameter selection and reported in the summary tables. For Tables 7 and 8 which are split by entity type, we use the model that had the highest F1 score on the validation set using the chosen hyperparameters. We conduct a grid search over hyperparameters of embedding size, number of layers, and hidden layer size for each neural network model, more details are given in the respective sections.

5.4 Feedforward Neural Network Model (FFNN-Neg)

Motivated by our analysis of the distribution negation markers can have around negated entities as demonstrated in Figure 2, we use a large symmetric input window for our feedforward network. As can be seen in Figure 5.4, in order to predict if token t_i is negated, we concatenate its context window of 15 tokens from the left and 15 tokens from the right and feed it to our network. If the input is shorter on either side, we pad it using a reserved *PAD* token that has its own embedding.

In order to encode entity information for the target token, we include a learned embedding that encodes the word token if the current input is not an entity, or the surrogate *ENT* token if it is an entity. After encoding the target token and its context, we concatenate the embeddings and feed them through a feedforward network with a hidden layer of dimensionality 128 and a ReLU non-linearity. We apply dropout of 0.5 to the input and the hidden activation and normalise the hidden activation using Layer Normalisation [Ba et al., 2016]. The feedforward network has a single output with a sigmoid activation for binary classification.

In addition to hyperparameter details already specified in the introduction, we set our embedding size to 100 and used a hidden layer with 128 hidden units and a single hidden layer, as we found these setting to give the best F1 score on the validation set. We use a batch size of $4,096$ context windows, which corresponds to a batch size of approximately 256 sentences assuming an average sentence length of 16 tokens. We train until no improvement on negation detection F1 score is obtained for 500 parameter updates on the held out split of the development set. From here on we refer to this model as FFNN-Neg.

5.5 LSTM Based Model (BiLSTM-Neg)

The BiLSTM-Neg model consists of a bidirectional LSTM followed by a fully connected layer. Two inputs were created for the model, a word embedding and a binary measure of whether the token is an entity. The target output is an array of binary outputs, one for each token, of whether the token should be negated. Models were built in Python using the PyTorch [Paszke et al., 2017] machine learning library. The Adam optimiser [Kingma and Ba, 2015] was used with a learning rate of $\alpha = 0.001$ and a weight decay of 0.0001. We apply a batch size of 32 documents.

The best model consisted of a two-layer LSTM of hidden size 128 using a 300 dimension word embedding to which a dropout of 0.5 has been applied. The final hidden state of the LSTM layer was concatenated with the binary entity measure to create the input for a fully connected layer, after which a sigmoid activation was applied to create the output to which threshold of 0.5 is applied to turn the float values into a binary measure. Negation predictions for entity tokens are combined with the entity types (without negation information) from the manually annotated reports for the final output. From here on this model is referred to as BiLSTM-Neg.

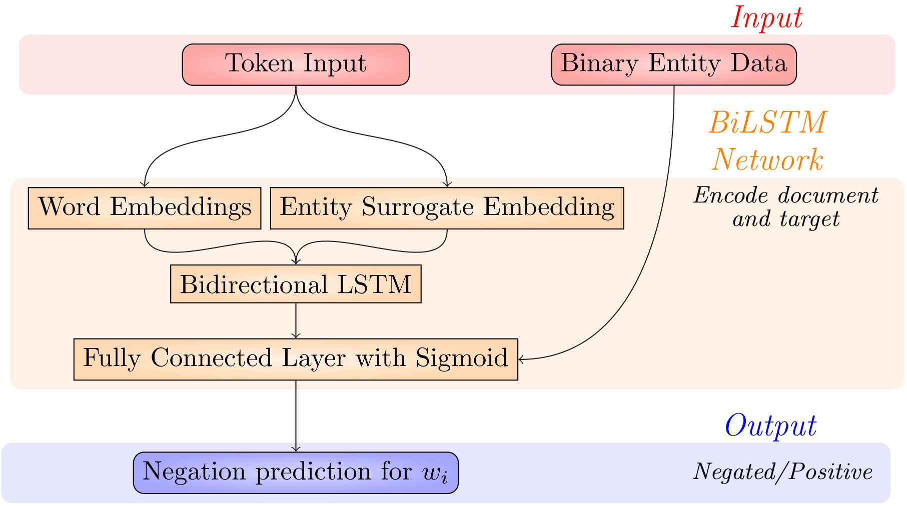

Input
Token Input
Binary Entity Data
BiLSTM Network
Encode document and target
Word Embeddings
Entity Surrogate Embedding
Bidirectional LSTM
Fully Connected Layer with Sigmoid
Output
Negation prediction for w_i
Negated/Positive

Fig. 5. Token data is input for a word embedding and entity surrogate embedding, to which Dropout of 0.5 is applied. These are used as input to a bidirectional LSTM. The hidden state of the LSTM is concatenated with Binary Entity Data for input into a fully connected layer with sigmoid activation for negation prediction.

6 Evaluation and Results

All five models described above were run over the two test sets (ESS test and Tayside test) with the manually annotated word and entity data as input (without negation attributes for entities). The negated entity output for each algorithm and the manually annotated entities (with negation specified) were then converted to BIO ConLL format to evaluate using the CoNLL [Tjong Kim Sang, 2002] perl evaluation script.[3] This is used for computing precision, recall and F1 score both as a total and per entity type for diseases and modifiers separately. Negative entities were taken as the positive class. We report 95% confidence intervals for precision and recall of the rule-based models and the mean ± 1 standard deviation over 5 runs with different random seeds for the machine learning models.

Table 4 shows the results for negated entities on the ESS Test dataset for each of the models. The rule-based model, EdIE-R-Neg, and the two machine learning models, BiLSTM-Neg and FFNN-Neg, perform similarly, achieving F1 scores within a point. Each model performed better in precision than recall, likely due to a larger number of non-negated entities in the dataset. However, while the confidence intervals for precision and recall for the rule-based model overlap with one standard deviation from the mean of the machine learning models, meaning we lack confidence of the superiority of the EdIE-R-Neg model in these individual aspects, the F1 score for EdIE-R-Neg is higher than one standard deviation above the

[3] `https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt`

Model	Precision	Recall	F1 Score
Rule-Based			
EdIE-R-Neg	98.02 (96.92–98.72)	97.71 (96.56–98.48)	**97.86**
pyConText	91.52 (89.61–93.09)	94.28 (92.62–95.57)	92.88
NegBio	93.90 (92.09–95.32)	83.35 (80.86–85.57)	88.31
Machine Learning			
FFNN-Neg	97.55 (97.45-97.65)	96.96 (96.76-97.16)	97.25 (97.17-97.34)
BiLSTM-Neg	97.45 (96.81-98.08)	96.69 (96.33-97.05)	97.07 (96.77-97.36)

Table 4. Overall Negated Entity (n. 961) results on the ESS test set. We report precision (positive predictive value), recall (sensitivity) and F1 score (harmonic mean of precision and recall) for negative annotations as well as 95% confidence intervals for precision and recall in parentheses for Rule Based systems and the mean ±1 standard deviation of 5 runs with different random seeds for Machine Learning models.

mean for both machine learning models, indicating EdIE-R-Neg will be the higher performing model for the majority, if not all, runs of the machine learning models.

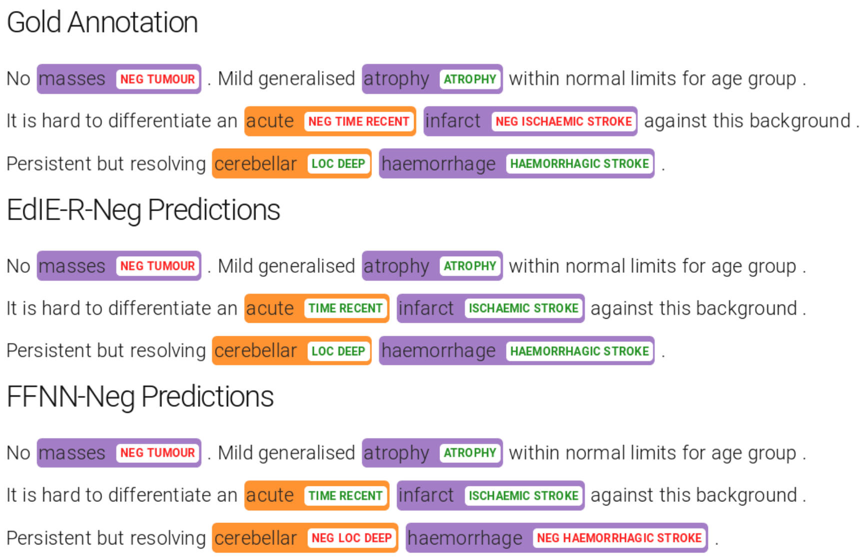
Of the machine learning networks, FFNN-Neg looks to have the higher performance with higher mean scores for precision, recall, and subsequently F1 score. However, this is inconclusive due to the mean scores of FFNN-Neg falling within one standard deviation of BiLSTM-Neg scores, meaning for any one run there is a large chance of either model having higher performance.

As expected, due to being developed on different datasets, the two existing rule-based methods, pyConText and NegBio, had weaker performance than those developed on the ESS development data, with NegBio particularly struggling with recall. pyConText shows a bias towards negated entities leading to a higher recall than precision, indicating that its negation rules are too broad for this dataset.

Largely the misclassifications on the ESS test set made by the custom-built models, EdIE-R-Neg, FFNN-Neg and BiLSTM-Neg, were very similar, but there are some key differences. BiLSTM makes a number of errors for the modifier entity of *time recent*, missing more of the negated instances than FFNN-Neg, while EdIE-R-Neg makes less errors than both. Additionally, the machine learning models perform worse than EdIE-R-Neg on each of the other modifier entities. An example of misclassifications made is shown in Figure 6.

There were a few key entities where NegBio made most of its misclassifications, spread across disease and modifier entities. PyConText followed a more similar pattern of error to our models, but greater in number, particularly for modifier entities.

Table 5 shows the results for negated entities on the Tayside test set. Similar to the ESS test set EdIE-R-Neg outperforms the other approaches in precision and F1 score, however for recall the margin is small between all three approaches, with FFNN-Neg scoring marginally higher and the confidence interval for EdIE-R-Neg overlapping with one standard deviation from the mean of the machine learning



Gold Annotation

No masses NEG TUMOUR . Mild generalised atrophy ATROPHY within normal limits for age group .

It is hard to differentiate an acute NEG TIME RECENT infarct NEG ISCHAEMIC STROKE against this background .

Persistent but resolving cerebellar LOC DEEP haemorrhage HAEMORRHAGIC STROKE .

EdIE-R-Neg Predictions

No masses NEG TUMOUR . Mild generalised atrophy ATROPHY within normal limits for age group .

It is hard to differentiate an acute TIME RECENT infarct ISCHAEMIC STROKE against this background .

Persistent but resolving cerebellar LOC DEEP haemorrhage HAEMORRHAGIC STROKE .

FFNN-Neg Predictions

No masses NEG TUMOUR . Mild generalised atrophy ATROPHY within normal limits for age group .

It is hard to differentiate an acute TIME RECENT infarct ISCHAEMIC STROKE against this background .

Persistent but resolving cerebellar NEG LOC DEEP haemorrhage NEG HAEMORRHAGIC STROKE .

Fig. 6. Example negation detection output of EdIE-R-Neg and FFNN-Neg on synthetic data. Diseases are represented in purple and modifiers in orange. Negation is marked as *NEG* in the tag. Both systems get the first two sentences correct. Correctly tagging the third sentence relies on understanding that "hard to differentiate x" implies that x is likely not present, something both EdIE-R-Neg and FFNN-Neg fail. FFNN-Neg gets the tags of the last sentence wrong presumably by relying on the presence of *but* or information that this is a haemorrhagic stroke entity, which is more commonly negated than not.

models meaning we lack confidence of any model's superiority on any one run of the machine learning models.

In contrast to the ESS test set, BiLSTM-Neg achieves a higher F1 score than FFNN-Neg due to a stronger precision, however we draw no definite conclusion of superiority as one standard deviation below the mean for BiLSTM-Neg F1 score is lower than one standard deviation above the mean for FFNN-Neg F1 score, meaning there is a reasonable chance of either model having higher true performance. Additionally, FFNN-Neg scored higher than BiLSTM-Neg on recall, though the margin is small.

Both machine learning models perform better on recall than precision, indicating a bias towards over-prediction of negated entities, similar to the last sentence in the example in Figure 6. The decrease in precision led to a drop on F1 score and overall performance when compared to the performance on the ESS test set. The cross-training experiments below describe how this drop is reduced through addition of Tayside development data.

NegBio has a more balanced performance in contrast to the ESS test set, while pyConText continues to show a bias towards negated entities with a greater gap between a high recall and low precision compared to results on ESS test.

Model	Precision	Recall	F1 Score
Rule-Based			
EdIE-R-Neg	98.31 (97.35–98.93)	98.68 (97.80–99.21)	**98.50**
pyConText	89.81 (87.92–91.43)	96.90 (95.68–97.78)	93.22
NegBio	89.81 (87.83–91.49)	88.63 (86.58–90.40)	89.21
Machine Learning			
FFNN-Neg	93.67 (92.42–94.93)	98.80 (98.64–98.96)	96.16 (95.57–96.75)
BiLSTM-Neg	95.28 (94.32–96.25)	98.52 (98.24–98.79)	96.87 (96.41–97.33)

Table 5. Overall Negated Entity (n. 988) results on the Tayside test set. We report precision (positive predictive value), recall (sensitivity) and F1 score (harmonic mean of precision and recall) for negative annotations as well as 95% confidence intervals for precision and recall in parentheses for Rule Based systems and the mean ±1 standard deviation of 5 runs with different random seeds for Machine Learning models.

Model	Precision	Recall	F1 Score
Train on ESS			
FFNN-Neg	93.67 (92.42–94.93)	98.80 (98.64–98.96)	96.16 (95.57–96.75)
BiLSTM-Neg	95.28 (94.32–96.25)	98.52 (98.24–98.79)	96.87 (96.41–97.33)
Train on ESS +20% Tayside			
FFNN-Neg	95.06 (93.93–96.19)	98.78 (98.44–99.12)	96.88 (96.41–97.35)
BiLSTM-Neg	96.11 (95.74–96.49)	98.52 (98.25–98.78)	97.30 (97.12–97.48)
Train on ESS +100% Tayside			
FFNN-Neg	94.26 (93.14–95.38)	**99.14** (98.97–99.30)	96.63 (96.11–97.16)
BiLSTM-Neg	**97.25** (96.87–97.62)	98.12 (97.71–98.53)	**97.68** (97.63–97.73)

Table 6. Effect of training on additional Tayside data on reported Tayside test scores (transfer learning). The scores reported are the mean over 5 runs with different random initialisation seeds and the range reported in parentheses is a standard deviation below and above the mean.

FFNN-Neg and BiLSTM-Neg followed a similar pattern of errors on the Tayside test set, with the exception of FFNN-Neg making a much greater number of errors on modifier entities, in particular *location deep*.

As on the ESS test set, pyConText makes misclassifications largely on the same entity types as the machine learning models, but with a greater number of errors on modifier entities. NegBio made higher numbers of misclassifications on a number of entities, however a high proportion of its errors were made on the same entity, *location deep*, as FFNN-Neg made a high proportion of its errors on.

As expected due to training on data more similar to the test set, Table 6 shows that adding Tayside data to the ESS training data (transfer learning) increases the F1 scores of the machine learning models on the Tayside test set, particularly for Bi-LSTM-Neg which consistently improved on adding 20% and then 100% of Tayside development data.

The improvement for BiLSTM-Neg came from increases in precision (ESS only: 95.28; +20% Tayside: 96.11; +100% Tayside: 97.25) where previously both machine learning models were weak, giving a more equal balance between precision and recall. While recall did dip slightly from 98.52 to 98.12 for BiLSTM-Neg on adding 100%, this was a smaller drop and largely the high recall was maintained.

A drop in precision from 95.06 with 20% Tayside added to 94.26 with 100% Tayside added for FFNN-Neg led to a fall in F1 score from 96.88 to 96.63, despite recall increasing from 98.78 to 99.14. This is in contrast to the behaviour of BiLSTM-Neg (ESS only: 95.28; +20% Tayside: 96.11; +100% Tayside: 97.25) and the behaviour of FFNN-Neg on adding 20% Tayside (ESS only: 93.67; +20% Tayside: 95.06; +100% Tayside: 94.26), which improved upon a weakness in precision. Despite the increases in F1 score both models remain weaker in performance than EdIE-R-Neg (EdIE-R-Neg: 98.50; BiLSTM-Neg: 97.68; FFNN-Neg: 96.88).

7 Discussion and Further Work

Through summing the true and false counts in Tables 7 and 8 in section A in the Appendix, we see differences in total counts of entities between the ESS and Tayside datasets, such as for metastatic tumour (ESS Test Set: 12; Tayside Test Set: 119), as well as ratios of negated to non-negated entities (Ischaemic Stroke - ESS Test Set: 149:316; Tayside Test Set: 176:130). The difference is largely due to ESS being reports from scans conducted as part of a Stroke study, while Tayside are routine scans. This may account for the imbalance that occurs between precision and recall on the Tayside test set for the machine learning models when training solely on ESS development data.

Negation patterns between the ESS and Tayside datasets are likely to be similar, for reasons including their source and the annotation process, therefore it is unsurprising that methods developed on the ESS dataset, such as EdIE-R-Neg, score higher in F1 score on the Tayside test set than the existing rule-based systems developed on other datasets, pyConText and NegBio. PyConText and NegBio were also developed on clinical reports, and their lower performance compared to our models on the ESS and Tayside datasets indicates that even within the same domain generalisation is difficult.

We have shown that introducing a small amount of the novel dataset, Tayside, into training data can improve performance on an unseen subset of the Tayside dataset, increasing precision and reducing the deficit in overall performance to EdIE-R-Neg. As discussed negation patterns for the ESS and Tayside datasets are likely to be similar, and we hypothesise that on a novel dataset where the negation patterns differ more greatly to the development dataset that adaption of the neural network method will be quicker and more effective compared to a rule-based system. Future work will aim to test the hypothesis that a neural network pretrained on ESS development data and fine-tuned on the novel dataset would outperform EdIE-

R-Neg. The i2b2 clinical dataset [Uzuner et al., 2011] is a strong candidate for the novel dataset.

In this work we eliminated the time needed to tune hyperparameters for adaption to the Tayside dataset by using the same model architecture selected from experiments on the ESS dataset only, and future work could further reduce the time taken to adapt to a new dataset through only fine-tuning the fully connected top layer of BiLSTM-Neg, or the final layer of FFNN-Neg, instead of the whole network.

Another common way to increase performance of natural language models is to use pretrained embeddings from much larger datasets. Pretrained word embeddings were tested in early versions of our neural network models without much success, and other work has also demonstrated inconclusive performance gains using either word or contextual embeddings that are not domain specific and instead trained on unrelated datasets [Cornegruta et al., 2016; Alsentzer et al., 2019]. However, our datasets will grow as we receive more reports and pretraining embeddings on the larger dataset is worth revisiting, as well as fine-tuning contextual domain specific embeddings from the recently available clinical BERT [Alsentzer et al., 2019].

Additionally, the gap in performance between EdIE-R-Neg and the machine learning models could be closed further through addition of a couple of key rules which could be explored in further work. Using rules, such as regular expressions like *(no evidence of — no evidence of developing) entity*, in post-processing has been found to increase performance over either method separately [Peng et al., 2019]. Similar rules tailored to the errors made could be used instead of the preprocessing outlined in the following paragraph to address where the machine learning approaches struggle with modifier entities.

Ensemble based algorithms use multiple systems to obtain better predictive performance than any single constituent system on it's own [Polikar, 2006], and are another way performance could be increased. However, due to similarities in performance by entity type and sub-type, the specific errors made by the systems may overlap significantly, and so improvements might be marginal for the approaches and datasets in this paper.

To further reduce the numbers of misclassifications made by our models, it may be useful to target certain entities. Modifier entities like *time recent* and *location deep* had more errors than other entities, and further work could investigate preprocessing methods that may give more information on these entities to the machine learning models.

From the literature we were expecting NegBio to outperform pyConText [Peng et al., 2017]. The weaker performance of NegBio came from a lower recall, indicating that non-negated entities were over-predicted, possibly due to negation cues in our dataset not present in that in which NegBio was developed. This is in contrast to pyConText which over-predicted negation. That a few key entities provided most of the errors for NegBio indicates there might be some key rules missing that are needed for the ESS and Tayside datasets more than other datasets.

8 Conclusion

We illustrate that negation detection, when conceptualised as a binary problem of presence vs non-presence, is a task that is relatively straightforward when working with radiology reports. We demonstrate this is the case, by showing that both rule-based, specifically ones optimised for a dataset, and neural network approaches, can perform highly accurately. The lower complexity of the task is also shown by the high performance of a relatively simple feedforward network, along with using training examples numbering only in the thousands.

Both machine learning alternatives to the rule-based EdIE-R-Neg proved effective, achieving very similar F1 Score performance on the ESS test set (EdIE-R-Neg: 97.86; BiLSTM-Neg: 97.07; FFNN-Neg: 97.25), with EdIE-R-Neg overall performing strongest. The pattern was similar for the Tayside test set (EdIE-R-Neg: 98.50; BiLSTM-Neg: 96.87; FFNN-Neg: 96.16), however the gap between the machine learning models and EdIE-R-Neg was larger. On our datasets all three models outperformed the two baseline existing rule-based models, pyConText and NegBio, demonstrating the effectiveness of custom-built models for specific datasets and indicating the difficulty in generalisation between medical datasets.

The performance deficit of the machine learning models on the Tayside test set to EdIE-R-Neg was reduced through addition of Tayside data to the training data of the neural network models (BiLSTM-Neg: 97.68; FFNN-Neg: 96.88). BiLSTM particularly benefited from the additional data, largely in precision.

9 Funding

The research reported on in this paper was funded by the MRC Mental Health Data Pathfinder Award (MRC - MC_PC_17209). Moreover, Alex and Grover have been supported by the Alan Turing Institute via Turing Fellowships (EPSRC grant EP/N510129/1). Whiteley was supported by an MRC Clinician Scientist Award (G0902303) and is supported by a Scottish Senior Clinical Fellowship (CAF/17/01). Sudlow is Chief Scientist of UK Biobank and Director of Health Data Research UK Scotland. We would like to thank our funders for their support.

Additionally, we are grateful to The Health Informatics Centre (HIC), University of Dundee for providing anonymised brain imaging reports from NHS Tayside.

10 Availability of Data and Software

The annotated ESS corpus used for this research was created with funding from the Wellcome Trust. It is available for research uses on application to Prof. Cathie Sudlow (email: `Cathie.SudlowATed.ac.uk`) to bona fide researchers with a clear analysis plan, in line with the Wellcome Trust policy on data-sharing.[4] We are in the process of creating a release of EdIE-R free for research purposes.[5] For more information contact Dr. Beatrice Alex (email: `balexATed.ac.uk`).

[4] `wellcome.ac.uk/what-we-do/topics/data-sharing`

[5] `https://www.ltg.ed.ac.uk/software/edie-r/`

References

Alex, B., Grover, C., Tobin, R., Sudlow, C., Mair, G., and Whiteley, W. (2019). Text mining brain imaging reports. *Journal of Biomedical Semantics*.

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jindi, D., Naumann, T., and McDermott, M. (2019). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization.

Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Chapman, W., Bridewell, W., Hanbury, P., F. Cooper, G., and Buchanan, B. (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34:301–310.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Cornegruta, S., Bakewell, R., Withey, S., and Montana, G. (2016). Modelling radiological language with bidirectional long short-term memory networks. *CoRR*, abs/1609.08409.

Cruz, N. P., Taboada, M., and Mitkov, R. (2017). A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*, 67(9):2118–2136.

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

Gorinski, P. J., Wu, H., Grover, C., Tobin, R., Talbot, C., Whalley, H., Sudlow, C., Whiteley, W., and Alex, B. (2019). Named Entity Recognition for Electronic Health Records: A Comparison of Rule-based and Machine Learning Approaches. *arXiv e-prints*, page arXiv:1903.03985.

Goryachev, S., Sordo, M., Zeng, Q. T., and Ngo, L. (2006). Implementation and evaluation of four different methods of negation detection. *Boston, MA: DSG*.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Grover, C. and Tobin, R. (2006). Rule-based chunking and reusability. In *Proceedings of LREC 2006*, pages 873–878.

Harkema, H., Dowling, J. N., Thornblade, T., and Chapman, W. W. (2009). Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839 – 851. Biomedical Natural Language Processing.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Horng, S., Sontag, D. A., Halpern, Y., Jernite, Y., Shapiro, N. I., and Nathanson,

L. A. (2017). Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. *PLOS ONE*, 12(4):1–16.

Hripcsak, G. and Rothschild, A. S. (2005). Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298.

Huang, Y. and Lowe, H. (2007). A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American Medical Informatics Association : JAMIA*, 14:304–11.

Jackson, C., Crossland, L., Dennis, M., Wardlaw, J., and Sudlow, C. (2008). Assessing the impact of the requirement for explicit consent in a hospital-based stroke study. *QJM: Monthly Journal of the Association of Physicians*, 101(4):281–289.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Maldonado, R., Goodwin, T., and M Harabagiu, S. (2017). Active deep learning-based annotation of electroencephalography reports for cohort identification. In *CRI*, volume 2017, pages 229–238.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mehrabi, S., Krishnan, A., Sohn, S., Roch, A. M., Schmidt, H., Kesterson, J., Beesley, C., Dexter, P., Schmidt, C. M., Liu, H., and Palakal, M. (2015). Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of Biomedical Informatics*, 54:213 – 219.

Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L., and Jin, Z. (2016). How Transferable are Neural Networks in NLP Applications? *arXiv e-prints*, page arXiv:1603.06111.

Mutalik, P., Deshpande, A. M., and Nadkarni, P. M. (2001). Research paper: Use of general-purpose negation detection to augment concept indexing of medical documents: A quantitative study using the umls. *Journal of the American Medical Informatics Association : JAMIA*, 8 6:598–609.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Peng, Y., Wang, X., Lu, L., Bagheri, M., Summers, R., and Lu, Z. (2017). NegBio: a high-performance tool for negation and uncertainty detection in radiology reports. *arXiv e-prints*, page arXiv:1712.05898.

Peng, Y., Yan, K., Sandfort, V., Summers, R. M., and Lu, Z. (2019). A self-attention based deep learning method for lesion attribute detection from ct reports. *arXiv preprint arXiv:1904.13018*.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45.

Pons, E., Braun, L. M. M., Hunink, M. G. M., and Kors, J. A. (2016). Natural Language Processing in Radiology: A Systematic Review. *Radiology*, 279(2):329–343.

Pratt, L. Y., Mostow, J., and Kamm, C. A. (1991). Direct transfer of learned information among neural networks. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*, AAAI'91, page 584–589. AAAI Press.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Taylor, S. and Harabagiu, S. (2018). The role of a deep-learning method for negation detection in patient cohort identification from electroencephalography reports. *Proceedings of the AMIA Annual Symposium*, 2018:1018–1027.

Tjong Kim Sang, E. F. (2002). Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.

Uzuner, Ö., South, B. R., Shen, S., and DuVall, S. L. (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *CoRR*, abs/1705.02315.

Wu, S., Miller, T., Masanz, J., Coarr, M., Halgrim, S., Carrell, D., and Clark, C. (2014). Negation's not solved: Generalizability versus optimizability in clinical natural language processing. *PLOS ONE*, 9(11):1–11.

A Result Tables Split by Entity

	Rule-Based												Machine Learning							
	pyConText				NegBio				EdIE-R-Neg				FFNN-Neg				BiLSTM-Neg			
Entity/Modifier	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
Atrophy	3	142	8	0	3	131	19	0	3	150	0	0	3	150	0	0	3	150	0	0
Haemorrhagic Stroke	212	47	4	4	194	48	3	22	212	50	1	4	212	49	2	4	213	49	2	3
Haemorrhagic Transf.	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Ischaemic Stroke	145	293	13	4	117	303	3	32	145	302	4	4	141	302	4	8	144	301	5	5
Meningioma Tumour	0	5	1	2	0	6	0	2	0	6	0	2	2	5	1	0	0	6	0	2
Metastatic Tumour	10	0	0	2	9	0	0	3	9	0	0	3	10	0	0	2	9	0	0	3
Microhaemorrhage	3	6	1	0	2	7	0	1	3	7	0	0	3	7	0	0	3	7	0	0
Small Vessel Disease	3	268	5	0	3	266	7	0	3	271	2	0	3	271	2	0	3	273	0	0
Stroke	4	22	0	0	4	22	0	0	4	22	0	0	4	21	1	0	4	20	2	0
Subarachnoid Haem.	2	7	1	0	1	7	1	1	2	8	0	0	2	8	0	0	2	8	0	0
Subdural Haematoma	99	9	0	1	76	9	0	24	100	9	0	0	100	9	0	0	100	9	0	0
Tumour	160	3	0	3	138	3	0	25	159	3	0	4	160	3	0	3	160	3	0	3
Location Cortical	8	396	8	0	7	403	1	1	8	402	2	0	6	403	1	2	8	401	3	0
Location Deep	2	328	11	2	3	327	12	1	3	339	0	1	2	338	1	2	2	337	2	2
Time Old	4	298	19	0	3	313	4	1	4	312	5	0	3	311	6	1	4	309	8	0
Time Recent	250	54	13	37	240	65	2	47	283	62	5	4	279	62	5	8	271	63	4	16
Total Entities	642	803	33	16	548	803	33	110	641	829	7	17	641	826	10	17	642	827	9	16
Total Modifiers	264	1076	51	39	253	1108	19	50	298	1115	12	5	290	1114	13	13	285	1110	17	18
Total	906	1879	84	55	801	1911	52	160	939	1944	19	22	931	1940	23	30	927	1937	26	34

Table 7. Comparison of results for all models on the ESS Test Set broken down by Disease and Modifier entities including total counts. We take negation to be the positive class, therefore TP (True Positives) are correctly predicted negated Disease and Modifier entities while TN (True Negatives) are correctly predicted non-negated Disease and Modifier entities.

	Rule-Based												Machine Learning							
	pyConText				NegBio				EdIE-R-Neg				FFNN-Neg				BiLSTM-Neg			
Entity/Modifier	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
Atrophy	1	166	1	0	0	166	1	1	1	167	0	0	1	166	1	0	1	167	0	0
Glioma Tumour	0	9	0	0	0	8	1	0	0	9	0	0	0	7	2	0	0	9	0	0
Haemorrhagic Stroke	255	36	3	0	247	37	2	8	254	39	0	1	255	38	1	0	255	37	2	0
Haemorrhagic Transf.	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
Ischaemic Stroke	174	127	3	2	157	130	0	19	173	130	0	3	173	128	2	3	172	126	4	4
Meningioma Tumour	0	2	0	0	0	2	0	0	0	2	0	0	0	1	1	0	0	2	0	0
Metastatic Tumour	56	55	7	1	45	62	0	12	55	61	1	2	52	59	3	5	53	61	1	4
Microhaemorrhage	3	3	0	0	2	3	0	1	3	3	0	0	3	3	0	0	3	3	0	0
Small Vessel Disease	2	168	3	0	1	169	2	1	2	171	0	0	2	166	5	0	2	170	1	0
Stroke	6	3	0	0	6	3	0	0	5	3	0	1	6	3	0	0	6	3	0	0
Subarachnoid Haem.	6	10	0	0	4	10	0	2	4	10	0	2	5	10	0	1	4	10	0	2
Subdural Haematoma	79	10	1	5	59	9	2	25	83	10	1	1	84	10	1	0	84	10	1	0
Tumour	252	44	2	5	240	46	0	17	253	46	0	4	255	44	2	2	254	44	2	3
Location Cortical	4	459	13	0	2	470	2	2	4	472	0	0	4	471	1	0	4	470	2	0
Location Deep	9	520	45	0	7	481	84	2	9	563	2	0	9	523	42	0	9	557	8	0
Time Old	11	118	29	0	6	137	10	5	11	136	11	0	11	136	11	0	11	135	12	0
Time Recent	173	74	10	20	167	81	3	26	193	81	3	0	192	78	6	1	191	79	5	2
Total Entities	834	634	20	13	761	646	8	86	833	652	2	14	836	636	18	11	834	643	11	13
Total Modifiers	197	1171	97	20	182	1169	99	35	217	1252	16	0	216	1208	60	1	215	1241	27	2
Total	1031	1805	117	33	943	1815	107	121	1050	1904	18	14	1052	1844	78	12	1049	1884	38	15

Table 8. Comparison of results for all models on the Tayside Test Set, as well as summary data for Disease and Modifier entities combined. We take negation to be the positive class, therefore TP (True Positives) are correctly predicted negated Disease and Modifier entities while TN (True Negatives) are correctly predicted non-negated Disease and Modifier entities.