Analyzing Patient Records to Establish If and When a Patient Suffered from a Medical Condition

James Cogley, Nicola Stokes, Joe Carthy and John Dunnion
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland
James.Cogley@ucdconnect.ie {Nicola.Stokes, Joe.Carthy, John.Dunnion}@ucd.ie

Abstract

The growth of digital clinical data has raised questions as to how best to leverage this data to aid the world of healthcare. Promising application areas include Information Retrieval and Question-Answering systems. Such systems require an in-depth understanding of the texts that are processed. One aspect of this understanding is knowing if a medical condition outlined in a patient record is recent, or if it occurred in the past. As well as this, patient records often discuss other individuals such as family members. This presents a second problem - determining if a medical condition is experienced by the patient described in the report or some other individual. In this paper, we investigate the suitability of a machine learning (ML) based system for resolving these tasks on a previously unexplored collection of *Patient History and Physical Examination reports*. Our results show that our novel Score-based feature approach outperforms the standard Linguistic and Contextual features described in the related literature. Specifically, near-perfect performance is achieved in resolving *if* a patient experienced a condition. While for the task of establishing *when* a patient experienced a condition, our ML system significantly outperforms the ConText system (87% versus 69% f-score, respectively).

1 Introduction

The growth of the digitization of clinical documents has fostered interest in how to best leverage this data in providing assistance in the world of healthcare, including novel information retrieval (Voorhees and Tong, 2010), question answering (Demner-Fushman and Lin, 2007; Patrick and Li, 2011) and clinical summarization systems (Feblowitz et al., 2011).

Given the richness of the language found in clinical reports, novel systems require a deeper understanding of this textual data. One aspect of this understanding is the *assertion status* of medical conditions (Demner-Fushman et al., 2011). The assertion status of a medical condition may refer to *Negation Resolution*, *Temporal Grounding* (deciding if a condition is currently or historical, and *Condition Attribution* (deciding if a condition is experienced by the patient described in the report or some other individual). The focus of this paper rests on the latter two tasks of *Temporal Grounding* and *Condition Attribution* as Negation has been satisfactorily addressed in Chapman et al. (2007).

Several approaches, ranging in complexity, have been proposed for resolving temporal information. Hripcsak et al. (2005) modeled the task as a constraint satisfaction problem. Another rule-based approach that achieved moderate results uses regular expressions matching occurrences of trigger terms (Chapman et al. 2007). A trigger term in this context refers to a term or phrase that provides strong evidence supporting the attribution (e.g., patient, family member) or temporality (e.g., current, past) of a condition. Given the limitations of solely using pre-composed trigger term lists, recent focus has been placed on the use of rule-based learning systems with different feature sets (Mowery et al., 2009). Section headers, tense and aspect are investigated as features, with promising results for the temporality task achieving an accuracy score of 89%. However, the authors' acknowledge that conclusions drawn must be tentative as a majority class classifier achieved an accuracy of 86.9% (only 13% of conditions in the dataset are historical).

Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012), pages 38–46,
Montréal, Canada, June 8, 2012.

This paper extends current work in the domain in the following ways. The dataset used in these experiments is generated from a collection of previously unannotated History & Physical (H&P) Examination reports. Prior work has focused on other report types such as discharge summaries and emergency department reports. In these cases the distribution of historical and recent conditions is often heavily skewed in favour of descriptions of recent conditions experienced by the patient. As H&P reports aim to provide a comprehensive picture of a patient's past and present state, a more uniform distribution of historical and recent conditions is present in this report type. This work extends previous work by exploring the use of machine learning (ML) as an alternative to hand-crafted rule based systems and rule-based ML approaches to resolving these tasks.

In this work, a comparative analysis of several ML algorithms from different paradigms are evaluated, in order to determine the best approach for our tasks. Building on this, the performance of four automatically extracted feature sets are evaluated, identifying their contributions and also their interactions. This work also extends existing work by automatically extracting features that were previously extracted manually as well as the proposal of a set of novel score-based features. The performance of the ML algorithms are compared to the rule-based system - ConText. Our results show that the ML approaches significantly outperform this rule-based system on the *Temporal Grounding* task.

2 Related Work

Natural Language Processing techniques have been shown to have many different uses in Clinical Text Analysis, such as in the representation (Sager et al., 1994) and understanding (Christensen, 2002) of clinical narratives, and frequently now in the context of more elaborate large-scale systems, such as a clinical decision support system (Demner-Fushman et al., 2009).

Given the sensitive nature of clinical narratives and the difficulty in obtaining data collections for experimental purposes, evaluation and comparison of NLP systems in this domain is difficult. However, recently anonymised data provided by the *Biomedical Language Understanding (BLU) Lab* at the University of Pittsburgh as well as datasets provided as part of the i2b2/VA 2010 challenge (Uzuner et al., 2011), has greatly aided NLP research into the processing of clinical narratives. The dataset provided by BLU Lab and used in this work consists of 101,711 reports of several different report types ranging from discharge summaries to surgical pathology reports. The report types differ in content, technical language and structure. For example, surgical pathology reports are very technical and explicit in the information that they convey, e.g. results of a biopsy, blood cell counts etc. In contrast, discharge summaries and consultation reports are more expressive, and aim to create a more complete patient profile, e.g. including work and personal circumstances. The BLU Lab have published a number of papers on the topic of resolving the assertion status of medical conditions (Chapman et al., 2007; Harkema et al., 2009; Mowery et al., 2009). Their `ConText` algorithm (Chapman et al., 2007) uses handcrafted regular expressions, along with trigger terms and termination terms to determine characteristics of a condition mention in a text. The condition characteristics investigated included negation, temporality (recent, historical, hypothetical) and experiencer (patient, other). Their approach worked very well on the negation and hypothetical temporality, achieving an f-score of 97% in determining negation and an f-score of 88% in resolving hypothetical conditions. However, the approach was less successful when determining historical conditions and their experiencer, with f-scores of 71% and 67%, respectively. These results were generated on emergency room reports only.

In more recent work, their algorithm was applied to 5 other types of clinical document, namely: surgical pathology, operative procedure, radiology, echocardiogram and discharge summaries (Harkema et al., 2009). Results achieved on these new datasets were largely the same, with f-scores for negation ranging between 75% and 95%, and for hypothetical conditions ranging between 76% and 96%. Again, a marked drop-off was seen for historical conditions, with few occurrences of conditions for other experiencers annotated in the datasets (i.e. relatives) making it difficult to draw definitive conclusions from this work.

Although this manual rule based approach has performed well and is advocated due to its ease of implementation (Meystre et al., 2008), Harkema et al. (2009) note its limitations in resolving historical conditions, and encourage the possibility of statistical classifiers in which information other than lexical items, are considered as features. Further work investigating the use of Machine Learning (Uzuner et al., 2009; Mowery et al., 2009) has seen posi-

tive breakthroughs in resolving the assertion status of medical conditions.

The 2010 i2b2 challenge (Uzuner et al., 2011) provided a rigid and standardized platform for evaluating systems in resolving the assertion status of medical conditions found in Discharge Summaries. The challenge consisted of three subtasks:*Concept Extraction*, *Assertion* and *Relation Identification*. The second subtask of Assertion involved the development of systems that resolved the assertion status of medical conditions. As part of the assertion task there were six possible assertion statuses: present, absent, uncertain, conditional, or not associated with the patient. Systems submitted to this challenge ranged from more simplistic pattern matching techniques to more complex supervised and semi-supervised approaches (de Bruijn et al., 2011; Clark et al., 2011). The datasets used in the 2010 i2b2 challenge were not available to non-participants at the time the experiments presented in this work were conducted. We plan to explore these datasets in future work. This research investigates patient vs. non-patient conditions as well as past vs. present conditions in order to fine tune feature-sets that may be generalized to further assertion statuses.

In the context of this paper, while the BLU Lab clinical report collection is available, the medical condition annotations are not. As already stated, our intention is to explore a machine learning approach to these tasks. For this purpose we annotated a portion of the consultation report section of the collection. There were two reasons for this - firstly, the BLU Lab have not reported results on this report type so there is no duplication of annotation effort and secondly, it turns out that the consultation reports are a much richer source of historical conditions and condition attribution than any of the report types annotated previously.

3 Method

3.1 Corpus

For this task, 120 H&P reports were randomly extracted from the BluLab's NLP repository. As already stated, this report type's fuller descriptions make it richer than previous datasets in instances of condition attribution and temporal grounding. A breakdown in the distributions of these annotations can be seen in Tables 1 and 2.

H&P reports may vary in the organization of content, but the content is mostly uniform, expressing the same information about patients (Sager et al., 1987). As well as this, many reports feature headings for different sections of the report (*past medical history*, *impression*), information which can be used as features in a classification task. Before annotating conditions found in the text, preprocessing was required in order to retain such information.

Table 1: Annotated Condition Attribution Occurrences

Class	Count
Patient	872
Other	93
Total	965

Table 2: Annotated Temporal Grounding Occurrences

Class	Count
Historical	448
Recent	424
Total	872

3.1.1 Preprocessing

Preprocessing of the data consisted of a simple Java program that extended Lingpipe[1] tools in order to correctly split sentences on this dataset, and extract the heading for the section in which the sentence is contained.

The preprocessing outputs the sentence number, followed by a separator, the sentence's contents and the heading under which the sentence features. Sentences were split for ease of annotation and also to allow parsing and part-of-speech tagging by the C&C[2] parsing tools. C&C was chosen for its scalabilty, speed and the accuracy of its biomedical language models. A cursory analysis of its output indicates that its performance is acceptable. As C&C does not provide a sentence splitter, Lingpipe's splitter was availed of for this task.

3.1.2 Annotation

Annotation of the dataset was performed by two annotators over a 60 hour period. The inter-annotator agreement was measured using the kappa statistic (Carletta, 1996). A kappa statistic of 0.78 was achieved. The annotators were presented with the collection, to annotate with an XML like tag "*CONDITION*". This tag must have two attributes, "*EXP*" representing condition attribution and "*HIST*"

[1] http://alias-i.com/lingpipe/

[2] http://svn.ask.it.usyd.edu.au/trac/candc

representing the temporal grounding of the condition.

- *HIST*: A value of 1 indicates the occurrence of a historical condition, where 0 describes a current or recent condition. e.g. *"The patient presented with <CONDITION NUM="1" HIST="0"> renal failure </CONDITION>"* would indicate the condition "renal failure" is current.
- *EXP*: A value of 1 implies the experiencer is the patient with 0 signifying "other". e.g. *"The patient has a family history of <CONDITION NUM="1" EXP="0">hypertension </CONDITION>"* signifies the condition "hypertension" is not experienced by the patient.

3.2 Machine Learning Algorithms

Early work in the resolution of assertion status primarily focused on the use of manually created rule-based systems, with more recent work focusing on statistical and ML methods. However, the domain of ML contains many sub-paradigms and varying approaches to classification. In this paper, three ML methods that have not been previously applied to this task are explored. These three classifiers, namely Naive Bayes, k-Nearest Neighbour and Random Forest represent the paradigms of probabilistic, lazy and ensemble learning, respectively.

Naive Bayes is a probabilistic classifier implementing Bayes Theorem. As a result, features implemented using this classifier are deemed to be independent. Despite this strong assumption it has been shown to be more than successful in text classification tasks such as spam filtering (Provost, 1999).

k-Nearest Neighbour (kNN) (Cover and Hart, 1967) is a simple pattern recognition algorithm that classifies an instance according to its distance to the *k* closest training instances. This algorithm has been chosen to represent the paradigm of lazy learning, i.e. there is no training phase as all computation is performed at the classification stage. Despite its simplicity, k-NN has often produce high accuracy results in comparison to other approaches (Caruana, 2006).

The final classifier chosen for this task represents the state-of-the-art in machine learning, namely the Random Forest algorithm (Breiman, 2001). A Random Forest consists of many different decision trees, combining bagging (Breiman, 1996), and random feature selection.

3.3 Features

In this section, a list of features contributing to this task are presented. All features are automatically extracted using a set of tools developed by the authors. Section 3.3.1 presents score-based features that are unique to this work. Section 3.3.2 describes the implementation of features outlined in Chapman et al (2007). Section 3.3.3 and Section 3.3.4 present features similar to those investigated in Mowery et al. (2009).

3.3.1 Score based features

Scored based features used in this system extend and reinforce Trigger List features by computing a normalized score for the number of occurrences of Trigger List terms[3]. This feature aims to add further granularity to the decision making of the ML algorithms, presenting a floating point number rather than a binary one.

The equation for computing these scores is defined as follows.

$$s = \frac{N_t}{(N_w - S_w)} \quad (1)$$

N_t represents the number of trigger terms found in the sentence that contains the condition, N_w is the total number of words in the sentence, with S_w being the number of stopwords[4]. These scores are calculated for each type of trigger term. For example, for trigger type *relative_mention*, a score is calculated using mentions of relatives in the sentence.

3.3.2 Trigger List Features

- `precededByHistTerm`: This feature performs a look-up for trigger terms from the historical word list, checking if it directly precedes the condition. An example historical trigger term would be "history of" as in "a history of diabetes". If a condition, such as diabetes, is modified by a historical trigger term, it will return 1, otherwise 0.
- `containsHistMention`: This is a weaker form of `precededByHistTerm`, checking simply if a trigger term from the historical list occurs in the same sentence as the condition. If one does, it will return 1 otherwise 0.
- `hasRelativeMention`: If the sentence which contains the condition also contains a trigger

[3]These trigger lists may be downloaded at `http://csserver.ucd.ie/~jcogley/downloads/wordlists.tar.gz`

[4]The list of stopwords may be downloaded at `http://csserver.ucd.ie/~jcogley/downloads/stopwords.txt`

term from the experiencer list such as 'mother', 'father' or 'uncle' it will return 1, otherwise 0.

- `hasPatientMention`: 1 if the sentence mentions the patient, otherwise 0.
- `containsDeath`: 1 if it contains the terms "deceased", "died" from the death trigger terms list otherwise 0. A sentence describing death is more likely to refer to a relative, rather than the patient.
- `mentionsCommunity`: 1 if one of "area", "community" from the geographical trigger list is mentioned, otherwise 0. If a sentence describes a community, as in "there has been a recent outbreak of flu in the area", it is not referring to the patient, therefore the condition should not be attributed to the patient.
- `precededByWith`: 1 if the condition is directly preceded by "with", otherwise 0. "with" was found to have higher frequency when describing patients rather than individuals other than the patient. e.g. "Patient presented with high blood pressure and fever."
- `containsPseudoTerms`: Pseudo-historical terms or phrases may mention a term that is found in the Historical list, but do not indicate that a condition mention in the same sentence is being used in a historical context. For example, "poor history" is a pseudo-historical trigger term. It uses a historical trigger term ("history"); however "poor history" refers to the incomplete nature of the patient's medical history, not the historical nature of their condition. This feature returns 1 on the occurrence of a pseudo trigger term, otherwise 0.

3.3.3 Contextual features

In resolving the textual context of conditions, it is important to look at what surrounds the condition beyond the lexical items. With these contextual features, we can capture that section in which a sentence occurs, and how many conditions occur in the sentence.

- `isInFamHist`: The importance of header information is motivated by the assumption that conditions that fall under explicit headings, are more than likely to have a greater affinity to the heading. This feature returns 1 if it is under *Family History*, 0 otherwise.
- `isInList`: A binary feature denoting whether a condition occurs as part of a list of conditions, with one condition per line. Returns 1 if it is a member of such a list, otherwise 0.
- `numOfConditions`: This feature represents the number of conditions present in a given sentence. A higher number of conditions indicates that the condition may be part of a list. Sentences that contain a list of conditions tend to list past conditions rather than recently suffered illnesses.

3.3.4 Linguistically motivated features

Three features were designed to monitor the effect of the verb tense on a condition. This feature has already been shown to aid the classification process (Mowery et al., 2009). For this task, linguistic features were extracted from the output of the C&C parsing tool, using both part-of-speech tags along with dependency information.

- `hasPastTense`: A binary feature with 1 indicating the sentence contains a past tense verb, 0 otherwise. e.g. "The patient previously suffered renal failure" would return 1. If a condition is modified by a past tense verb, it has occurred in the past.
- `hasPresentTense`: A binary feature with 1 indicating the sentence contains a present tense verb, 0 otherwise. If a condition is modified by a present tense verb, the condition is current. e.g. "the patient presents coughing".
- `containsModalVerb`: A binary feature with 1 indicating the sentence contains a modal verb, 0 otherwise. e.g. "palpitations may have been caused by anxiety". The presence of the modal "may" following the condition indicates the condition is currently being examined and is therefore recent.
- `tenseInClause`: Analyzes the tense found in the same syntactic clause as the condition being examined. For example, in "abdominal pain has ceased, but patient now complains of lower extremity pain", "abdominal pain" has a past tense within its clausal boundary, where the clause which contains "lower extremity pain" has a present tense verb.
- `tenseChange`: Determines whether the verb tense used in the clause that contains the condition differs with the verb in another clause in the sentence. e.g. "Migraines persist yet palpitations resolved". This feature allows finer granularity in resolving the tense surrounding conditions, such as the description of current conditions in the context of the patient's history.

4 Experiment Setup & Evaluation

There are two aims of the experiments reported in this section: firstly, to evaluate the performance of ML algorithms in resolving the assertion status of medical conditions. Secondly, to assess the performance of individual feature sets in order to discover the most contributory and informatory features or sets of features. To evaluate the latter, classifications using all possible combinations of feature sets (listed in Table 3) were performed.

Table 3: Feature-set Combinations

ID	Feature-Sets
TrigLingConScore	trigger, linguistic, score-based, contextual
TrigLingScore	trigger, linguistic, score-based
TrigLingCon	trigger, linguistic, contextual
TrigConScore	trigger, score-based, contextual
LingConScore	linguistic, score-based, contextual
TrigLing	trigger, linguistic
TrigScore	trigger, score-based
TrigCon	trigger, contextual
LingScore	linguistic, score-based
LingCon	linguistic, contextual
ConScore	score-based, contextual
Trigger	trigger
Ling	linguistic
Score	score-based
Con	contextual

4.1 Evaluation

The systems are evaluated by the metrics precision, recall and f-score:

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

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

$$f = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives.

Systems are evaluated using both cross-validation and hold-out methods. In the hold-out method there are two data sets, one used for training the classifier and a second for testing it on a blind sub-set of test material. 10-fold cross-validation is performed on the training sets and final results are reported in this paper on the held-out blind test set. Three hold-out classification splits were experimented with (i.e., train/test splits: 30%/70%; 50%/50%; 70%/30%). We found that results for each of the data splits and cross-validation experiments were largely uniform. To avoid repetition of results, Section 5 focuses on experiments using a held-out method training/test split of 70%/30%. All hold-out experiments were implemented using Weka's (Hall et al., 2009) Experimenter interface. Cross-Validation experiments were performed using a script developed by the authors in conjunction with Weka's API. This allowed the ML approaches and the ConText algorithm to be evaluated against the same test-folds.

4.1.1 Comparison with a rule-based system

ConText (Chapman et al., 2007) is a simple yet effective rule-based system designed to resolve the assertion status of medical conditions. Comparative analysis is performed between an implementation of ConText[5] and the ML approaches described in 3.2. The ML systems were trained on 70% of the dataset (610 conditions). The remaining 30% (262 conditions) was used as a test set for both ConText and the Machine Learning systems. For cross-validation experiments, ConText was evaluated against each test set fold. For the Condition Attribution experiments training was performed on 675 conditions with testing performed on 290 conditions. In evaluating Temporal Grounding the training set comprised of 610 conditions with the test-set containing 262 conditions.

5 Experimental Results

This section reports results of the experiments outlined in Section 4.

5.1 Condition Attribution

In a system that resolves the assertion status of medical conditions, it is of benefit to know who is experiencing the medical condition before resolving more complex information such as temporality. In this section, preliminary results on Condition Attribution are presented. The dataset used in evaluating the effectiveness of Condition Attribution was highly skewed, as shown in Table 1. This is a natural skew caused simply by the fact that reports discuss the patient more than other individuals (e.g., blood relatives). As a result the baseline score using a Majority Class classifier achieved an f-score of 95% (Table 4). Given these results, the contextual feature set contributes most, as shown by the removal of the contextual feature set in TrigLingScore coinciding with a drop in performance. However, the skewed dataset resulted in no statistical significance

[5] http://code.google.com/p/negex/downloads/detail?name=GeneralConText.Java.v.1.0_10272010.zip

between classifiers and feature-sets.

Table 4: Selected feature-sets (f-score) using Cross-Validation for the Condition Attribution task

ID	RFor	kNN	NB	Maj.
TrigLingConScore	100%	100%	100%	95%
TrigLingScore	96%	96%	96%	95%
TrigConScore	100%	100%	100%	95%
Con	100%	100%	100%	95%

In this task, ConText achieved an f-score of 99%. As there is little difference in scores achieved between ConText and the approaches in Table 4 - a manual rule-based system can be considered adequate for this task.

Taking a closer look at the performance of the feature sets, we see that the contextual feature set contributes most to the task with the removal of contextual features coinciding with a drop in performance e.g., TrigLingScore in Table 4. The strength of this feature set lies with the feature `isInFamHist`. This feature simply checks if the condition occurs under the heading "Family History". Its highly influential performance would indicate that its quite rare for the mention of another individual anywhere else in a clinical report. The Con run, which is solely composed of contextual features achieves near perfect performance, an indication that the contribution of other features to the task of Condition Attribution is minimal. Although this work used only H&P reports, this finding may be generalized to other report types such as Discharge Summaries which also explicitly mark sections pertaining to other individuals.

5.2 Temporal Grounding

The distribution of past and recent medical conditions is not skewed (as in the Condition Attribution task), and hence it presents a more challenging classification task. Despite the varying performance of individual classifiers and feature sets the results obtained are again largely similar across cross-validation and hold-out methods, with the performance of each training set fitting the distribution in Figure 1. Initial experiments investigated the use of another state-of-the-art classifier, the Support Vector Machine using a polykernel, however due to its relatively poor performance (70% f-score, with its precision soundly beaten by other approaches) it will not be discussed in further detail.

Random Forest proved to be the most effective classifier across almost all feature sets. However, kNN was a very near second place - Random Forest only significantly[6] outperformed kNN on two occasions (TrigLingConScore, LingConScore). In constrast, Naive Bayes performed poorly - despite having outperformed all other systems on the precision metric, it failed to outperform the baseline majority classifier on the recall.

Although the precision of ConText matches that of the Random Forest and kNN (Table 5), it is also let down by its recall performance. As a result, there is a statistical significant difference between its f-score and that of the Random Forest and kNN.

Table 5: Temporal Grounding ConText Comparison

System	Precision	Recall	F-score
`kNN`	80%	80%	80%
`RandomForest`	82%	84%	83%
`NaiveBayes`	91%	61%	72%
`ConText`	80%	61%	69%
`Majority`	55%	100%	71%

6 Discussion

The distribution of recent and historical conditions for the task of Temporal Grounding is more evenly distributed than that used in Condition Attribution, allowing for a more interesting comparison of the approaches and features employed.

Figure 1 shows the performance of each ML for each feature-set combination. Random Forest was expectedly the best performing algorithm. However, more surprising was the comparative performance of the often overlooked kNN algorithm. Both approaches statistically significantly outperformed the rule-based system ConText. Though the rule based system matched the high performing ML systems in terms of precision, it was significantly outperformed with respect to recall.

The most contributory feature set in the ML runs was the novel score-based feature set. This feature creates a normalized score for the occurrence of trigger terms in the same sentence as the medical condition in question. It was designed to reinforce the importance of trigger terms, by providing a numeric score to support the binary Trigger List feature. The addition of score-based features to any of the feature combinations coincided with a statistical significant boost in performance, with Score (composed solely of score-based features) outperforming half of all other feature combinations as seen in Figure 1,.

On the contrary, the effect of contextual features on the performance of the algorithms for Temporal

[6]Significance calculated by Paired T-Test with 95% confidence.

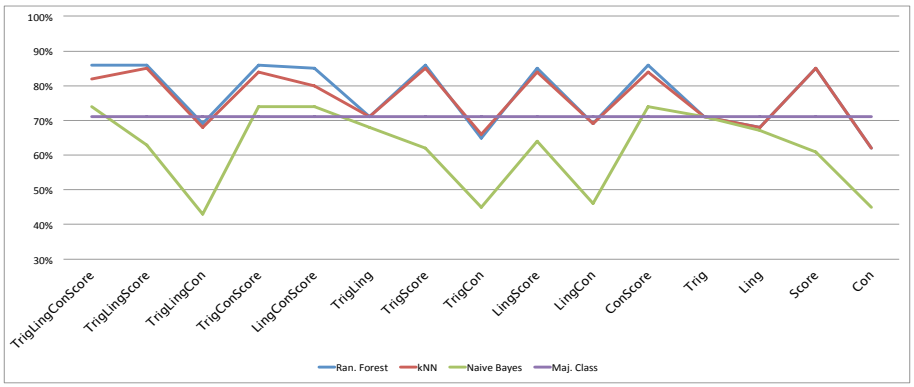

Figure 1: Temporal Grounding f-score performance with 70% Training Data

Grounding is minimal, or even detrimental to the task. For example, in Figure 1, the performance of the kNN algorithm increases from TrigLingConScore to TrigLingScore with the removal of contextual features. The performance of the Random Forest is unaffected by such detrimental features as it performs its own feature selection prior to classification. Though there are several feature set combinations reaching a high level of performance, the most effective approach combines trigger list terms, linguistic and score based features with the Random Forest algorithm.

These experiments extend previous work by providing a systematic, automated approach to feature extraction for the purpose of ML approaches to Temporal Grounding. They also indicate the high performance and contribution of our novel score-based features. These features are not designed to solely classify instances found in H&P reports and can be applied to other clinical reports such as Discharge Summaries and Emergency Department reports. Previous work has involved the use of the latter mentioned report types. Unfortunately, given the terms-of-use of these datasets they could not be made available to authors to facilitate comparative experiments.

7 Conclusion

In this paper, we proposed the use of machine learning (ML) in resolving if and when a patient experienced a medical condition. The implemented ML algorithms made use of features comprising of trigger terms, linguistic and contextual information, and novel score-based features. In an evaluation of these feature sets it was found that score-based features contributed to the task of resolving when a patient experienced a medical condition.

The ML approaches were also evaluated against the rule-based system ConText on a new annotated dataset of History & Physical (H&P) Examination Reports. In this evaluation it was discovered that the task of resolving *if a condition was experienced by the patient* was adequately solved by the ConText system, achieving an f-score of 99%. Although, the ML approaches proposed achieved perfect performance, there is no statistical significance between the result sets. However, the more challenging task of *deciding when a patient experienced a medical condition* is deemed to be best suited to a ML approach, with the top performing classifier *Random Forest* achieving an f-score of 87%, significantly outperforming ConText which achieved 69% on the same dataset .

The results achieved in these tasks have paved the way for several avenues of future work. We believe that the performance of these tasks is now sufficiently accurate to justify their inclusion in an Information Retrieval (IR) application. It is our intention to use our medical condition analysis techniques to annotate clinical documents and build an advanced IR system capable of taking advantage of this mark up in the context of the TREC Medical Records Track 2012[7]. With the availability of datasets such as that of the i2b2 Shared Task 2010 data, further work will include experimentation on these datasets as well as an investigation into further assertion statuses.

8 Acknowledgments

We are grateful to Dr Martina Naughton for her advice on many aspects of this paper. We also wish to acknowledge the support of Science Foundation Ireland, who fund this research under grant number 10/RFP/CMS2836.

[7] `http://groups.google.com/group/trec-med`

References

L. Breiman. 1996. Bagging predictors. *Machine Learning*, 24:123–140.

L. Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249 – 254.

R. Caruana. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of 23rd International Conference on Machine Learning*.

W. W. Chapman, D. Chu, and J. N. Dowling. 2007. Context: An algorithm for identifying contextual features from clinical text. In *BioNLP 2007: Biological, translational, and clinical language processing*, pages 81–88, June.

L. M. Christensen. 2002. Mplus: A probabilistic medical language understanding system. In *Proceedings of Workshop on Natural Language Processing in the Biomedical Domain*, pages 29–36.

C. Clark, J. Aberdeen, M. Coarr, D. Tresner-Kirsch, B. Wellner, A. Yeh, and L. Hirschman. 2011. Mitre system for clinical assertion status classification. *Journal of the American Medical Informatics Association*.

T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *Transactions on Information Theory*.

B. de Bruijn, C. Cherry, S. Kiritchenko, J. Martin, and X. Zhu. 2011. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association*.

D. Demner-Fushman and J. Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. In *Computational Linguistics*, pages 63–103.

D Demner-Fushman, W W. Chapman, and C J. McDonald. 2009. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42:760–772.

D. Demner-Fushman, S. Abhyankar, A. Jimeno-Yepes, R. Loane, B. Rance, F. Lang, N. Ide, E. Apostolova, and A. R. Aronson. 2011. A knowledge-based approach to medical records retrieval. In *TREC 2011 Working Notes*.

J. Feblowitz, A. Wright, H. Singh, L. Samal, and D. Sittig. 2011. Summarization of clinical information: A conceptual model. *Biomedical Informatics*.

M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explorations*.

H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman. 2009. Context: An algorithm for identifying contextual features from clinical text. *Journal of Biomedical Informatics*, 42(5):839–851.

G. Hripcsak, L. Zhou, S. Parsons, A. K. Das, and S. B. Johnson. 2005. Modeling electronic discharge summaries as a simple temporal constraint satisfaction problem. *Journal of the American Medical Informatics Association*, 12(1):55–63, January.

S. M. Meystre, G. K. Savova, K. C. Kipper-Schuler, and J. F. Hurdle. 2008. Extracting information from textual documents in the electronic health record: a review of recent research. *IMIA Yearbook of Medical Informatics*, pages 128–144.

D. L. Mowery, H Harkema, J. N. Dowling, J. L. Lustgarten, and W. W. Chapman. 2009. Distinguishing historical from current problems in clinical reports-which textual features help? In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processin*, pages 10–18. Association for Computational Linguistics.

J. Patrick and M. Li. 2011. An ontolotgy for clinical questions about the contents of patients notes. *Journal of Biomedical Informatics*.

J. Provost. 1999. Naive-bayes vs. rule-learning in classification of email. Technical report, The University of Texas at Austin.

N. Sager, C. Friedman, and M.S. Lyman. 1987. *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley.

N. Sager, M. Lyman, C. Bucknall, N. Nhan, and L. J. Tick. 1994. Natural language processing and the representation of clinical data. *Journal of the American Medical Informatics Association*, 1:142–160.

O. Uzuner, X. Zhang, and T. Sibanda. 2009. Machine learning and rule-based approaches to assertion classification. *Journal of the American Medical Informatics Association*, 16(1):109–115.

Ö. Uzuner, BR. South, S. Shen, and SL. DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*.

E. Voorhees and R. Tong. 2010. Overview of the trec 2011 medical records track. preprint.