ConText: An Algorithm for Identifying Contextual Features from Clinical Text

Wendy W. Chapman, David Chu, John N. Dowling

Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA
chapman@cbmi.pitt.edu

Abstract

Applications using automatically indexed clinical conditions must account for contextual features such as whether a condition is negated, historical or hypothetical, or experienced by someone other than the patient. We developed and evaluated an algorithm called ConText, an extension of the NegEx negation algorithm, which relies on trigger terms, pseudo-trigger terms, and termination terms for identifying the values of three contextual features. In spite of its simplicity, ConText performed well at identifying negation and hypothetical status. ConText performed moderately at identifying whether a condition was experienced by someone other than the patient and whether the condition occurred historically.

1 Introduction

Natural language processing (NLP) techniques can extract variables from free-text clinical records important for medical informatics applications performing decision support, quality assurance, and biosurveillance [1-6]. Many applications have focused on identifying individual clinical conditions in textual records, which is the first step in making the conditions available to computerized applications. However, identifying individual instances of clinical conditions is not sufficient for many medical informatics tasks—the context surrounding the condition is crucial for integrating the information within the text to determine the clinical state of a patient.

For instance, it is important to understand whether a condition is affirmed or negated, acute or chronic, or mentioned hypothetically. We refer to these as contextual features, because the information is not usually contained in the lexical representation of the clinical condition itself but in the context surrounding the clinical condition. We developed an algorithm called ConText for identifying three contextual features relevant for biosurveillance from emergency department (ED) reports and evaluated its performance compared to physician annotation of the features.

2 Background

2.1 Encoding Contextual Information from Clinical Texts

NLP systems designed to encode detailed information from clinical reports, such as MedLEE [1], MPLUS [7], and MedSyndikate [4], encode contextual features such as negation, uncertainty, change over time, and severity. Over the last ten years, several negation algorithms have been described in the literature [8-12]. Recently, researchers at Columbia University have categorized temporal expressions in clinical narrative text and evaluated a temporal constraint structure designed to model the temporal information for discharge summaries [13, 14].

ConText differs from most other work in this area by providing a stand-alone algorithm that can be integrated with any application that indexes clinical conditions from text.

2.2 Biosurveillance from ED Data

Biosurveillance and situational awareness are imperative research issues in today's world. State-of-the-art surveillance systems rely on chief complaints and ICD-9 codes, which provide limited clinical information and have been shown to perform with only fair to moderate sensitivity [15-18]. ED reports are a timely source of clinical informa-

BioNLP 2007: Biological, translational, and clinical language processing, pages 81–88,

tion that may be useful for syndromic surveillance. We are developing NLP-based methods for identifying clinical conditions from ED reports.

2.3 SySTR

We are developing an NLP application called SySTR (Syndromic Surveillance from Textual Records). It currently uses free-text descriptions of clinical conditions in ED reports to determine whether the patient has an acute lower respiratory syndrome. We previously identified 55 clinical conditions (e.g. cough, pneumonia, oxygen desaturation, wheezing) relevant for determining whether a patient has an acute lower respiratory condition [19]. SySTR identifies instances of these 55 clinical conditions in ED reports to determine if a patient has an acute lower respiratory syndrome. SySTR has four modules:

(1) Index each instance of the 55 clinical conditions in an ED report;
(2) For each indexed instance of a clinical condition, assign values to three contextual features;
(3) Integrate the information from indexed instances to determine whether each of the 55 conditions are *acute*, *chronic*, or *absent*;
(4) Use the values of the 55 conditions to determine whether a patient has an acute lower respiratory syndrome.

We built SySTR on top of an application called caTIES [20], which comprises a GATE pipeline of processing resources (http://gate.ac.uk/). Module 1 uses MetaMap [5] to index UMLS concepts in the text and then maps the UMLS concepts to the 55 clinical conditions. For instance, Module 1 would identify the clinical condition Dyspnea in the sentence "Patient presents with a 3 day history of shortness of breath." For each instance of the 55 conditions identified by Module 1, Module 2 assigns values to three contextual features: Negation (*negated, affirmed*); Temporality (*historical, recent, hypothetical*); and Experiencer (*patient, other*). For the sentence above, Module 2 would assign Dyspnea the following contextual features and their values: Negation—*affirmed*; Temporality—*recent*; Experiencer—*patient*. Module 3, as described in Chu and colleagues [21], resolves contradictions among multiple instances of clinical conditions, removes conditions not experienced by the patient, and assigns a final value of *acute*, *chronic*, or *absent* to each of the 55 conditions. Module 4 uses machine learning models to determine whether a patient has acute lower respiratory syndrome based on values of the conditions.

The objective of this study was to evaluate an algorithm for identifying the contextual information generated by Module 2.

3 Methods

We developed an algorithm called ConText for determining the values for three contextual features of a clinical condition: Negation, Temporality, and Experiencer. The same algorithm is applied to all three contextual features and is largely based on a regular expression algorithm for determining whether a condition is negated or not (NegEx [9]). ConText relies on trigger terms, pseudo-trigger terms, and scope termination terms that are specific to the type of contextual feature being identified. Below we describe the three contextual features addressed by the algorithm, details of how ConText works, and our evaluation of ConText.

3.1 Three Contextual Features

Determining whether a patient had an acute episode of a clinical condition, such as cough, potentially involves information described in the context of the clinical condition in the text. We performed a pilot study to learn which contextual features affected classification of 55 clinical conditions as *acute*, *chronic*, or *absent* [21]. The pilot study identified which contextual features were critical for our task and reduced the number of values we initially used.

The contextual features for each indexed clinical condition are assigned default values. ConText changes the values if the condition falls within the scope of a relevant trigger term. Below, we describe the contextual features (default values are in parentheses).

(1) **Negation** (*affirmed*): ConText determines whether a condition is negated, as in "No fever."
(2) **Temporality** (*recent*): ConText can change Temporality to *historical* or *hypothetical*. In its current implementation, *historical* is defined as beginning at least 14 days before the visit to the ED, but the algorithm can easily be modified to change the length of time. ConText would mark Fever in "Patient should return if she develops fever" as *hypothetical*.

(3) **Experiencer** (*patient*): ConText assigns conditions ascribed to someone other than the patient an Experiencer of *other*, as in “The patient’s father has a history of CHF.”

3.2 Contextual Feature Algorithm

As we examined how the contextual features were manifested in ED reports, we discovered similar patterns for all features and hypothesized that an existing negation algorithm, NegEx [9], may be applicable for all three features.

NegEx uses two regular expressions (RE) to determine whether an indexed condition is negated:

RE1: <trigger term> <5w> <indexed term>
RE2: <indexed term> <5w> <trigger term>

<5w> represents five words (a word can be a single word or a UMLS concept), and the text matched by this pattern is called the scope. NegEx relies on three types of terms to determine whether a condition is negated: trigger terms, pseudo-trigger terms, and termination terms. Trigger terms such as “no” and “denies” indicate that the clinical conditions that fall within the scope of the trigger term should be negated. Pseudo-trigger terms, such as “no increase,” contain a negation trigger term but do not indicate negation of a clinical concept. A termination term such as “but” can terminate the scope of the negation before the end of the window, as in “She denies headache but complains of dizziness.”

ConText is an expansion of NegEx. It relies on the same basic algorithm but applies different term lists and different windows of scope depending on the contextual feature being annotated.

3.3 ConText Term Lists

Each contextual feature has a unique set of trigger terms and pseudo-trigger terms, as shown in Table 1. The complete list of terms can be found at http://web.cbmi.pitt.edu/chapman/ConText.html. Most of the triggers apply to RE1, but a few (marked in table) apply to RE2. ConText assigns a default value to each feature, then changes that value if a clinical condition falls within the scope of a relevant trigger term.

Although trigger terms are unique to the contextual feature being identified, termination terms

Table 1. Examples of trigger and pseudo-trigger terms for the three contextual features. If all terms are not represented in the table, we indicate the number of terms used by ConText in parentheses.

<table>
<tr><th colspan="4">Temporality (default = recent)</th></tr>
<tr><th>Trigger terms for hypothetical</th><th>Pseudo-trigger terms</th><th>Trigger terms for historical</th><th>Pseudo-trigger terms (10)</th></tr>
<tr><td>if
return
should [he|she]
should there
should the patient
as needed
come back [for|to]</td><td>if negative</td><td>General triggers
history
previous^
History Section title^^
Temporal Measurement triggers^^^
<time> of
[for|over] the [last|past] <time>
since (last) [day-of-week|week|month|season|year]</td><td>history, physical
history taking
poor history
history and examination
history of present illness
social history
family history
sudden onset of</td></tr>
</table>

<table>
<tr><th colspan="2">Experiencer (default = patient)</th></tr>
<tr><th>Trigger terms for other (12)</th><th>Pseudo-trigger terms</th></tr>
<tr><td>father(’s)
mother(’s)
aunt(’s)</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Negation (default = affirmed)</th></tr>
<tr><th>Trigger terms for negated (125)</th><th>Pseudo-trigger terms (16)</th></tr>
<tr><td>no
not
denies
without</td><td>no increase
not extend
gram negative</td></tr>
</table>

^ the scope for “previous” only extends one term forward (e.g., “for previous headache”)
^^Currently the only history section title we use is PAST MEDICAL HISTORY.
^^^ <time> includes the following regular expression indicating a temporal quantification: x[-|space] [day(s)|hour(s)|week(s)|month(s)|year(s)]. x = any digit; words in brackets are disjunctions; items in parentheses are optional. The first two temporal measurement triggers are used with RE1; the third is used with RE2. For our current application, a condition lasting 14 days or more is considered *historical*.

may be common to multiple contextual features. For instance, a termination term indicating that the physician is speaking about the patient can indicate termination of scope for the features Temporality and Experiencer. In the sentence "History of COPD, presenting with shortness of breath," the trigger term "history" indicates that COPD is *historical*, but the term "presenting" terminates the scope of the temporality trigger term, because the physician is now describing the current patient visit. Therefore, the condition Dyspnea ("shortness of breath") should be classified as *recent*. Similarly, in the sentence "Mother has CHF and patient presents with chest pain," Experiencer for CHF should be *other*, but Experiencer for Chest Pain should be *patient*.

We compiled termination terms into conceptual groups, as shown in Table 2.

Table 2. ConText's termination terms. Column 1 lists the type of termination term, the number of terms used by Context, and the contextual feature values using that type of termination term. Column 2 gives examples of the terms.

Type of Term	Examples
Patient (5) Temporal (hypothetical) Experiencer (other)	Patient, who, his, her, patient's
Presentation (12) Temporal (historical) Experiencer (other)	Presents, presenting, complains, was found, states, reports, currently, today
Because (2) Temporal (hypothetical)	Since, because
Which (1) Experiencer (other)	Which
ED (2) Temporal (historical)	Emergency department, ED
But (8) Negation (negated)	But, however, yet, though, although, aside from

3.4 ConText Algorithm

The input to ConText is an ED report with instances of the 55 clinical concepts already indexed. For each clinical condition, ConText assigns values to the three contextual features. ConText's algorithm is as follows[1]:

[1] This algorithm applies to RE1. The algorithm for RE2 is the same, except that it works backwards from the trigger term and does not look for pseudo-trigger terms.

Go to first trigger term in sentence
If term is a pseudo-trigger term,
Skip to next trigger term
Determine scope of trigger term
If termination term within scope,
Terminate scope before termination term
Assign appropriate contextual feature value to all indexed clinical concepts within scope.

The scope of a trigger term depends on the contextual feature being classified. The default scope includes all text following the indexed condition until the end of the sentence. Thus, in the sentence "He should return for fever" the scope of the Temporality (hypothetical) trigger term "return" includes the segment "for fever," which includes an indexed condition Fever. The default scope is overridden in a few circumstances. First, as described above, the scope can be terminated by a relevant termination term. Second, if the trigger term is a <section title>, the scope extends throughout the entire section, which is defined previous to ConText's processing. Third, a trigger term itself can require a different scope. The Temporality (historical) term "previous" only extends one term forward in the sentence.

3.5 Evaluation

We evaluated ConText's ability to assign correct values to the three contextual features by comparing ConText's annotations with annotations made by a physician.

Setting and Subjects. The study was conducted on reports for patients presenting to the University of Pittsburgh Medical Center Presbyterian Hospital ED during 2002. The study was approved by the University of Pittsburgh's Institutional Review Board. We randomly selected 120 reports for patients with respiratory-related ICD-9 discharge diagnoses for manual annotation. For this study, we used 30 reports as a development set and 90 reports as a test set. In addition to the annotated development set, we used a separate set of 100 unannotated ED reports to informally validate our term lists.

Reference Standard. A physician board-certified in internal medicine and infectious diseases with 30 years of experience generated manual annotations for the development and test reports. He used GATE (http://gate.ac.uk/) to highlight every individual annotation in the text referring to any of the 55 clinical conditions. For every annotation, he assigned values to the three contextual features, as shown in Figure 1.

Previous experience in annotating the 55 conditions showed that a single physician was inadequate for generating a reliable reference standard [19]. The main mistake made by a single physician was not marking a concept that existed in the text. We used NLP-assisted review to improve physician annotations by comparing the single physician's annotations to those made by SySTR. The physician reviewed disagreements and made changes to his original annotations if he felt his original annotation was incorrect. A study by Meystre and Haug [22] used a similar NLP-assisted review methodology and showed that compared to a reference standard not using NLP-assisted review, their system had higher recall and the same precision.

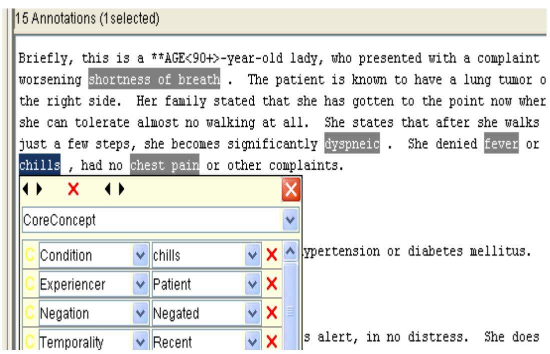

Figure 1. When the physician highlights text, GATE provides a drop-down menu to select the Clinical Condition and the values of the Contextual Features.

Outcome Measures. For each contextual feature assigned to an annotation, we compared ConText's value to the value assigned by the reference standard. We classified the feature as a true positive (TP) if ConText correctly changed the condition's default value and a true negative (TN) if ConText correctly left the default value. We then calculated recall and precision using the following formulas:

$$\text{Recall}: \frac{number\ of\ TP}{(number\ of\ TP + number\ of\ FN)}$$

$$\text{Precision}: \frac{number\ of\ TP}{(number\ of\ TP + number\ of\ FP)}$$

For the Temporality feature, we calculated recall and precision separately for the values *historical* and *hypothetical*. We calculated the 95% confidence intervals (CI) for all outcome measures.

4 Results

Using NLP-assisted review, the reference standard physician made several changes to his initial annotations. He indexed an additional 82 clinical conditions and changed the title of the clinical condition for 48 conditions, resulting in a total of 1,620 indexed clinical conditions in the 90 test reports. The reference standard physician also made 35 changes to Temporality values and 4 changes to Negation. The majority of Temporality changes were from *historical* to *recent* (17) and from *hypothetical* to *recent* (12).

Table 3 shows ConText's recall and precision values compared to the reference standard annotations. About half of the conditions were *negated* (773/1620). Fewer conditions were *historical* (95/1620), *hypothetical* (40/1620), or experienced by someone *other* than the patient (8/1620). In spite of low frequency for these contextual feature values, identifying them is critical to understanding a patient's current state. ConText performed best on Negation, with recall and precision above 97%. ConText performed well at assigning the Temporality value *hypothetical*, but less well on the Temporality value *historical*. Experiencer had a small sample size, making results difficult to interpret.

Table 3. Outcome measures for ConText on test set of 90 ED reports.

Feature	TP	TN	FP	FN	Recall 95% CI	Precision 95% CI
Negation	750	824	23	23	97.0 96-98	97.0 96-98
Temporality (*historical*)	66	1499	23	32	67.4 58-76	74.2 64-82
Temporality (*hypothetical*)	33	1578	2	7	82.5 68-91	94.3 81-98
Experiencer	4	1612	0	4	50.00 22-78	100 51-100

5 Discussion

We evaluated an extension of the NegEx algorithm for determining the values of two additional contextual features—Temporality and Experiencer. ConText performed with very high recall and precision when determining whether a condition was negated, and demonstrated moderate to high performance on the other features.

We performed an informal error analysis, which not only isolates ConText's errors but also points out future research directions in contextual feature identification.

5.1 Negation

ConText's negation identification performed substantially better than NegEx's published results [9], even though ConText is very similar to NegEx and uses the same trigger terms. Several possible explanations exist for this boost in performance. First, our study evaluated negation identification in ED reports, whereas the referenced study on NegEx applied to discharge summaries. Second, ConText only applied to 55 clinical conditions, rather than the large set of UMLS concepts in the NegEx study. Third, the conditions indexed by SySTR that act as input to ConText are sometimes negated or affirmed before ConText sees them. For some conditions, SySTR addresses internal negation in a word (e.g., "afebrile" is classified as Fever with the Negation value *negated*). Also, SySTR assigns Negation values to some conditions with numeric values, such as negating Tachycardia from "pulse rate 75." Fourth, ConText does not use NegEx's original scope of five words, but extends the scope to the end of the sentence. It would be useful to compare ConText's scope difference directly against NegEx to determine which scope assignment works better, but our results suggest the increased scope may work well for ED reports.

ConText's errors in assigning the Negation value were equally distributed between FN's and FP's (23 errors each). Some false negatives resulted from missing trigger terms (e.g., "denying"). Several false negatives resulted from the interaction between ConText and SySTR's mapping rules. For example, in the sentence "chest wall is without tenderness," SySTR maps the UMLS concepts for "chest wall" and "tenderness" to the condition Chest Wall Tenderness. In such a case, the negation trigger term "without" is caught between the two UMLS concepts. Therefore, RE1 does not match, and ConText does not change the default from *affirmed*. False positive negations resulted from our not integrating the rule described in NegEx that a concept preceded by a definite article should not be negated [23] (e.g., "has not been on steroids for his asthma") and from descriptions in the text whose Negation status is even difficult for humans to determine, such as "no vomiting without having the cough" and "patient does not know if she has a fever."

5.2 Temporality

Historical. ConText identified *historical* conditions with 67% sensitivity and 74% precision. Identifying historical conditions appears simple on the surface, but is a complex problem. The single trigger term "history" is used for many of the historical conditions, but the word "history" is a relative term that can indicate a history of years (as in "history of COPD") or of only a few days (as in "ENT: No history of nasal congestion"). The error analysis showed that ConText is missing trigger terms that act equivalently to the word "history" such as "in the past" ("has not been on steroids in the past for his asthma") and "pre-existing" ("pre-existing shortness of breath").

Some conditions that the reference standard classified as *historical* had no explicit trigger in the text, as in the sentence "When he sits up in bed, he develops pain in the chest." It may be useful to implement rules involving verb tense for these cases.

The most difficult cases for ConText were those with temporal measurement triggers. The few temporal quantifier patterns we used were fairly successful, but the test set contained multiple variations on those quantifiers, and a new dataset would probably introduce even more variations. For instance, ConText falsely classified Non-pleuritic Chest Pain as *historical* in "awoken at approximately 2:45 with chest pressure," because ConText's temporal quantifiers do not account for time of the day. Also, even though ConText's temporal quantifiers include the pattern "last x weeks," x represents a digit and thus didn't match the phrase "intermittent cough the last couple of weeks."

We were hoping that identifying historical conditions would not require detailed modeling of temporal information, but our results suggest otherwise. We will explore the temporal categories derived by Hripcsak and Zhou [13] for discharge summaries to expand ConText's ability to identify temporal measurement triggers.

Hypothetical. ConText demonstrated 83% recall and 94% precision when classifying a condition as *hypothetical* rather than *recent*. Again, missing trigger terms (e.g., "returning" and "look out for") and termination terms (e.g., "diagnosis") caused errors. The chief cause of false negatives was terminating the scope of a trigger term too early. For instance, in the sentence "She knows to return to the ED if she has anginal type chest discomfort which was discussed with her, shortness of breath, and peripheral edema" the scope of the trigger "return" was terminated by "her." The major limitation of regular expressions is evident in this example in which "her" is part of a relative clause modifying "chest discomfort," not "shortness of breath."

5.3 Experiencer

ConText's ability to identify an experiencer other than the patient suffered from low prevalence. In the test set of 90 reports, only 8 of the 1620 conditions were experienced by someone other than the patient, and ConText missed half of them. Two of the false negatives came from not including the trigger term "family history." A more difficult error to address is recognizing that bronchitis is experienced by someone other than the patient in "…due to the type of bronchitis that is currently being seen in the community." ConText made no false positive classifications for Experiencer.

5.4 Limitations and Future Work

Some of ConText's errors can be resolved by refining the trigger and termination terms. However, many of the erroneous classifications are due to complex syntax and semantics that cannot be handled by simple regular expressions. Determining the scope of trigger terms in sentences with relative clauses and coordinated conjunctions is especially difficult. We believe ConText's approach involving trigger terms, scope, and termination terms is still a reasonable model for this problem and hope to improve ConText's ability to identify scope with syntactic information.

A main limitation of our evaluation was the reference standard, which was comprised of a single physician. We used NLP-assisted review to increase the identification of clinical conditions and decrease noise in his classifications. It is possible that the NLP-assisted review biased the reference standard toward ConText's classifications, but the majority of changes made after NLP-assisted review involved indexing the clinical conditions, rather than changing the values of the contextual features. Moreover, most of the changes to contextual feature values involved a change in our annotation schema after the physician had completed his first round of annotations. Specifically, we allowed the physician to use the entire report to determine whether a condition was *historical*, which caused him to mark recent exacerbations of historical conditions as *historical*. A second physician is in the process of annotating the test set. The two physicians will come to consensus on their classifications in generating a new reference standard.

How good contextual feature identification has to be depends largely on the intended application. We tested SySTR's ability to determine whether the 55 clinical conditions were *acute*, *chronic*, or *absent* on a subset of 30 test reports [24]. SySTR made 51 classification errors, 22 of which were due to ConText's mistakes. In spite of the errors, SySTR demonstrated a kappa of 0.85 when compared to physician classifications, suggesting that because of redundancy in clinical reports, ConText's mistakes may not have a substantial adverse effect on SySTR's final output.

5.5 Conclusion

We evaluated a regular-expression-based algorithm for determining the status of three contextual features in ED reports and found that ConText performed very well at identifying negated conditions, fairly well at determining whether conditions were hypothetical or historical, and moderately well at determining whether a condition was experienced by someone other than the patient. ConText's algorithm is based on the negation algorithm NegEx, which is a frequently applied negation algorithm in biomedical informatics applications due to its simplicity, availability, and generalizability to various NLP applications. Simple algorithms for identifying contextual features of indexed conditions is important in medical language processing for improving the accuracy of information retrieval and extraction applications and for providing a baseline comparison for more sophisticated algorithms. ConText accepts any indexed clinical conditions as input and thus may be applicable to other NLP applications. We do not know how well ConText will perform on other report types, but see similar contextual features in discharge summaries, progress notes, and history and physical exams. Currently, ConText only identifies three contextual features, but we hope to extend the algorithm to other features in the future, such as whether a condition is mentioned as a radiology finding or as a diagnosis (e.g., Pneumonia).

Over and above negation identification, which can be addressed by NegEx or other algorithms, ConText could be useful for a variety of NLP tasks, including flagging historical findings and eliminating indexed conditions that are hypothetical or were not experienced by the patient. Ability to modify indexed conditions based on their contextual features can potentially improve precision in biosurveillance, real-time decision support, and information retrieval.

Acknowledgments. This work was supported by NLM grant K22 LM008301, "Natural language processing for respiratory surveillance."

References

1. Friedman C. A broad-coverage natural language processing system. Proc AMIA Symp 2000:270-4.
2. Fiszman M, Chapman WW, Aronsky D, Evans RS, Haug PJ. Automatic detection of acute bacterial pneumonia from chest X-ray reports. J Am Med Inform Assoc 2000;7(6):593-604.
3. Taira R, Bashyam V, Kangarloo H. A field theory approach to medical natural language processing. IEEE Transactions in Inform Techn in Biomedicine 2007;11(2).
4. Hahn U, Romacker M, Schulz S. MEDSYNDIKATE-a natural language system for the extraction of medical information from findings reports. Int J Med Inf 2002;67(1-3):63-74.
5. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp 2001:17-21.
6. Hazlehurst B, Frost HR, Sittig DF, Stevens VJ. MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record. J Am Med Inform Assoc 2005;12(5):517-29.
7. Christensen L, Haug PJ, Fiszman M. MPLUS: a probabilistic medical language understanding system. Proc Workshop on Natural Language Processing in the Biomedical Domain 2002:29-36.
8. Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J Am Med Inform Assoc 2001;8(6):598-609.
9. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 2001;34(5):301-10.
10. Elkin PL, Brown SH, Bauer BA, Husser CS, Carruth W, Bergstrom LR, et al. A controlled trial of automated classification of negation from clinical notes. BMC Med Inform Decis Mak 2005;5(1):13.
11. Herman T, Matters M, Walop W, Law B, Tong W, Liu F, et al. Concept negation in free text components of vaccine safety reports. AMIA Annu Symp Proc 2006:1122.
12. Huang Y, Lowe HJ. A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. J Am Med Inform Assoc 2007.
13. Hripcsak G, Zhou L, Parsons S, Das AK, Johnson SB. Modeling electronic discharge summaries as a simple temporal constraint satisfaction problem. J Am Med Inform Assoc 2005;12(1):55-63.
14. Zhou L, Melton GB, Parsons S, Hripcsak G. A temporal constraint structure for extracting temporal information from clinical narrative. J Biomed Inform 2005.
15. Chapman WW, Dowling JN, Wagner MM. Classification of emergency department chief complaints into seven syndromes: a retrospective analysis of 527,228 patients. Ann Emerg Med 2005;46(5):445-455.
16. Ivanov O, Wagner MM, Chapman WW, Olszewski RT. Accuracy of three classifiers of acute gastrointestinal syndrome for syndromic surveillance. Proc AMIA Symp 2002:345-9.
17. Chang HG, Cochrane DG, Tserenpuntsag B, Allegra JR, Smith PF. ICD9 as a surrogate for chart review in the validation of a chief complaint syndromic surveillance system. In: Syndromic Surveillance Conference Seattle, Washington; 2005.
18. Beitel AJ, Olson KL, Reis BY, Mandl KD. Use of emergency department chief complaint and diagnostic codes for identifying respiratory illness in a pediatric population. Pediatr Emerg Care 2004;20(6):355-60.
19. Chapman WW, Fiszman M, Dowling JN, Chapman BE, Rindflesch TC. Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap. Medinfo 2004;2004:487-91.
20. Mitchell KJ, Becich MJ, Berman JJ, Chapman WW, Gilbertson J, Gupta D, et al. Implementation and evaluation of a negation tagger in a pipeline-based system for information extraction from pathology reports. Medinfo 2004;2004:663-7.
21. Chu D, Dowling JN, Chapman WW. Evaluating the effectiveness of four contextual features in classifying annotated clinical conditions in emergency department reports. AMIA Annu Symp Proc 2006:141-5.
22. Meystre S, Haug PJ. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. J Biomed Inform 2006;39(6):589-99.
23. Goldin I, Chapman WW. Learning to detect negation with 'not' in medical texts. In: Proc Workshop on Text Analysis and Search for Bioinformatics at the 26th Annual International ACM SIGIR Conference (SIGIR-2003); 2003.
24. Chu D. Clinical feature extraction from emergency department reports for biosurveillance [Master's Thesis]. Pittsburgh: University of Pittsburgh; 2007.