Structuring Free-text Microbiology Culture Reports For Secondary Use

Wen-wai Yim1, Heather L. Evans, MD, MS2, Meliha Yetisgen, PhD1,3
1Biomedical and Health Informatics, School of Medicine, University of Washington, Seattle, WA; 2Department of Surgery, School of Medicine, University of Washington, Seattle, WA; 3Department of Linguistics, University of Washington, Seattle, WA

Abstract

Microbiology lab culture reports are a frequently used diagnostic tool for clinical providers. However, their incorporation into clinical surveillance applications and evidence-based medicine can be severely hindered by the free-text nature of these reports. In this work, we (1) created a microbiology culture template to structure free-text microbiology reports, (2) generated an annotated microbiology report corpus, and (3) built a microbiology information extraction system. Specifically, we combined rule-based, hybrid, and statistical techniques to extract microbiology entities and fill templates for structuring data. System performances were favorable, with entity f1-score 0.889 and relation f1-score 0.795. We plan to incorporate these extractions as features for our ongoing ventilator-associated pneumonia surveillance project, though this tool can be used as an upstream process in other applications. Our newly created corpus includes 1442 unique gram stain and culture microbiology reports generated from a cohort of 715 patients at the University of Washington Medical Facilities.

Background

Medical professionals frequently order microbiology laboratory culture tests to identify sources of bacterial infection, determine between differential diagnoses, and adjust antibiotic treatments. For providers, bacterial infections rates are an essential part of quality assurance as well as community outbreak awareness. However, although microbiology reports are semi-structured and electronically transmitted, the free-text nature of these reports provides a challenge to incorporating microbiology culture results in real-time surveillance applications. Unlike other medical diagnostic tests, microbiology culture results evolve over a period of time. Initially, samples of the patients’ blood, urine, or spinal fluid are collected and cultured to promote microorganism growth. These cultures may take days or weeks to grow before they can be reliably identified by microscope. In the meantime, only general appearance and concentration of bacterial colonies are available for clinicians to make decisions. After full identification, antibiotics are further tested against the microorganisms for drug resistance. Based on these outcomes, therapy options may also change. Therefore monitoring microbiology results is an important component in describing patients’ medical state. Furthermore, in surveillance or cohort-selection settings, this must be done automatically. Thus, the goal of our project is to build a microbiology results extraction system using natural language processing so that such results can be incorporated into automated systems.

Two previously published work on microbiology lab result identification and extraction came from the Salt Lake City Healthcare system Department of Veterans Affairs (VA) and the Vanderbilt VA system. The Salt Lake City VA system used a set of patterned rules to detect organisms, a quantity or concentration associated with it, and antibiotic susceptibilities. Evaluation was then performed by measuring successful identification of Staphylococcus aureus and methicillin-resistance. Meanwhile, the Vanderbilt VA system detected bacterial contamination in blood cultures using a hybrid NLP pipeline and Multi-Threaded Clinical Vocabulary Server (MCVS) NLP tools. They extract bacterium and asserted whether any were found. They then collected tuples of a bacterium, antibiotic, and sensitivity interpretations. Identification and evaluation were based on presence of all three elements in a tuple in a 24-hour period where less-specific references to a bacterium were removed.

Our system differs from the previous systems in that (a) we detected specimens from several sources including blood, (b) we identified microorganism, microorganism concentrations, antibiotics, antibiotic sensitivities, as well as other additional microbiology entities modifiers, and (c) we processed all reports and evaluated on entities and relations.
Methods

To build our system, we created, annotated, and evaluated on a clinical corpus from a cohort of 715 patients from the University of Washington (UW) Medical Center system. 13744 gram stain and culture report segments of microbiology reports were pulled from electronic medical records. The retrospective review of those reports was approved by the UW Human Subjects Committee of Institutional Review Board. Separate reports could be from the same patient or could describe the same specimen culture at different points in time. Duplicate reports were removed, resulting in a corpus of 1442 unique reports. We then built our system as an entity- and relation-extraction task. We report system performances for a five-fold cross-validation of the entire set.

Corpus Annotation

Our template scheme was designed with advice from a UW physician. Each template was represented as a composition of entities, spans of text with assigned label names, and relations, directed links between entities. In our task, entities captured identification, growth, antibiotic, and antibiotic susceptibility attributes, whereas relations ensured that the proper descriptions linked to the items they characterized. Figure 1 includes example reports annotated with entities and relations. A single template is the maximum set of entities connected by relations. One report can result in more than one filled microbiology template, as in Figure 1.B. We did not annotate entities that occurred as messages (e.g., Please inform the Micro Lab within 3 days of this organism report if susceptibility testing is necessary), advice (e.g., Rapidly growing yeasts are rarely a cause of pneumonia), or corrections (e.g., Previously reported as 3+Gram Positive Rods).

Our entities had the following definitions: (1) Organism: a microorganism found in the culture (e.g., bacteria, flora, fungus, yeast), (2) Organism quantity: the amount of the organisms in a culture (e.g., >10,000 col/ml, one colony, no, isolated), (3) Rating: a qualitative measurement of the amount of the organisms found in a culture (e.g., 1+, 2+, 3+, 4+), (4) Drug: a drug that was tested on the organism (e.g., penicillin), (5) Drug resistance: the susceptibility of the organism to the drug (e.g., susceptible, S, intermediate, I, resistant, R, no clsi interpretive criteria), (6) MIC: the minimum inhibitory concentration (MIC) or lowest concentration of an antimicrobial that inhibited the growth of a microorganism after overnight incubation (e.g., 2.0 Mcg/ml), (7) No Growth: instances of no organism growth (e.g., no growth), (8) No Growth Measurement: a time measurement of no growth (e.g., 2 days), (9) Specimen Description: a reference specimen (e.g., lower respiratory culture from endotracheal tube), (10) Specimen Date: the reference specimen date (e.g., same day), and (11) Reference Attribute: attributes that an outside specimen description points to (e.g., identification, sensitivities).

Our relations were defined as directed links between the two entity types as described following: (1) equivalentRefOf: organism to organism, or drug to drug, (2) hasQuantity: organism to organism quantity, (3) hasRating: organism to rating, (4) measuredBy: no growth to no growth measurement, (5) hasDrugDesc: organism to drug, (6) hasResistance: drug to drug resistance, or organism to drug resistance, (7) hasMIC: drug to MIC, (8) hasAttrRefIn: organism to specimen description, (9) timestamp: specimen description to specimen date, and (10) attr: specimen description to reference item.

Figure 1. Examples of the microbiology gram stain and culture reports annotated with entities and relations.
Using the brat rapid annotation tool\textsuperscript{6}, one medical student and one biomedical informatics graduate student annotated an initial set of 100 reports. The final annotator agreement, after one revision meeting, included an entity-level f1-measure of 0.964, a relation-level f1-measure of 0.937, and a template match f1-measure of 0.833, with 334 entities, 230 relations, and 85 templates exactly matched. Finally, our medical student annotator annotated the remaining 1342 reports of the corpus. The total number of entities and relations were 3720 and 2531, respectively.

**Entity and Relation Extraction**

We used the annotated corpus to train and evaluate an extraction system. Figure 2 presents the overall system architecture. Entity types were extracted independently using various strategies. Afterwards, all entities were consolidated and entity span overlap conflicts were resolved. Relations were identified from pair-wise classification of system entities.

**Entity Extraction**

Entities were identified by one of three strategies that exploit known characteristics about each type. For example, one group of entities was represented accurately with specific word patterns. These include entities of type ratings, whose value took on any in the set \{1+, 2+, 3+, 4+\}, and of type MIC, whose values were numeric with units mg/ml or µg/ml. A second group of entities, in contrast, could be identified by rules with very high recall but low precision. Specimen attributes for example could take the value “sensitivities” as in the phrase “Please see left lower lung 11/1 for sensitivities.” However, this entity also appeared in other un-annotated parts of reports such as in “Please contact the lab for additional sensitivities testing.”

The remaining entities did not have easily predictable patterns. These entities included organisms and specimen descriptions. Organism entities were difficult to define with rules because of diverse organism abbreviations, our definition of enzyme descriptions as organisms, as well as generic names or morphological descriptions defined as organism when cultures were not identifiable, (e.g. “colony no. 1”, “gram positive rods”, “polymorphonuclear cells”, “unidentified cells”). Specimen description used common medical terms (and their abbreviations) to refer to other specimens; however, likewise to organism entities, these inputs were not from a controlled vocabulary.

Thus, based on general characteristics of an entity type, the extraction process occurred in one of three ways: (1) rule-based extraction with regular expressions, (2) hybrid extraction, in which candidate entities were identified using high-recall rules, and then false positive entities were filtered out by using a machine learning classification algorithm, and (3) statistical extraction, which used a sequential classification approach.

Rules were crafted iteratively using regular expressions and controlled vocabularies to maximize recall for rule-based entities and hybrid candidate entities. To filter false positives from hybrid candidate entities, we trained a logistic regression classifier with LibLinear\textsuperscript{7}, set at l2-regularization and default parameters. Sequentially labeled entities were classified using a conditional random field tagging with CRFSuite\textsuperscript{8} using default parameters. Sequentially labeled entity features included n-grams, parts-of-speech, orthographic features, and highest-matching UMLS semantic types. During experimentation, we optimized for feature window sizes, feature sets, and sequential-label tag sets. Reports were tokenized using OpenNLP\textsuperscript{9} and UMLS medical concepts were identified by MetaMap\textsuperscript{10}. Because entities were independently extracted, it was possible for entities to have overlapping spans. We resolved such conflicts as follows: rule-based entities took the highest precedence, followed by hybrid entities, and statistical entities. For statistical entities that overlapped with rule-based or hybrid entities, we decided to truncate them until the statistical entity no longer overlapped with the other entity. When statistical or hybrid entities overlapped with the same type, we simply kept the entity with the highest learned confidence.

**Relation Extraction**

Relations were trained from entities in a pairwise fashion. Thus, for a given report all entity pair combinations were enumerated for classification. Because given the two entity types, it was possible to determine both direction and relation label, we simplified our task to a binary classification using logistic regression with LibLinear. Features were defined based on entity pairs, given all entity information in the report, and optimized during experimentation. Thus, some features included the labels and string-values of each entity pair, the number and type of entities...
between them, and n-grams around an entity pair. During post-processing, impossible relations, as defined by annotation guidelines, were removed. Equivalent relations were given a default direction to point from the earlier-appearing entity to the later-appearing entity.

Template Extraction and Evaluation

After extracting entities and relations, our system created templates for all connected extracted entities and relations. Our template match evaluation used the same calculation as our inter-annotator agreement, which was based on exact entity and relation match for each template.

Results

System performances for each entity extractor are outlined in Table 1. The f1-score ranges for rule-based, hybrid, and statistical extractors were 0.994-1.00, 0.857-0.962, and 0.800-0.827 respectively. Table 2 presents the micro results for all entity and relation extraction. Relation extraction with oracle entities (gold standard entities) is unsurprisingly much higher than those with system entities. Performance values for strict exact template match are shown in Table 3.

Table 1. Entity extraction performances. TP: True positive, FN: False negative, FP: False positive, P: Precision, R: Recall, F1: F1-score.

| Entity Type | Entity Label       | TP  | FN  | FP  | P   | R   | F1  |
|-------------|--------------------|-----|-----|-----|-----|-----|-----|
| Rule-based  | MIC                | 83  | 1   | 0   | 1.00| 0.988| 0.994|
|             | No Growth          | 26  | 0   | 0   | 1.00| 1.00 | 1.00 |
|             | Rating             | 453 | 0   | 3   | 0.993| 1.000| 0.997|
| Hybrid      | Drug               | 299 | 12  | 7   | 0.977| 0.961 | 0.969|
|             | Drug resistance    | 252 | 11  | 9   | 0.966| 0.958 | 0.962|
|             | No growth measure  | 24  | 2   | 1   | 0.960| 0.923 | 0.941|
|             | Organism quantity  | 637 | 119 | 94  | 0.871| 0.843 | 0.857|
|             | Reference item     | 128 | 6   | 6   | 0.955| 0.955 | 0.955|
|             | Specimen date      | 117 | 10  | 10  | 0.921| 0.921 | 0.921|
| Statistical | Organism           | 1133| 271 | 203 | 0.848| 0.807 | 0.827|
|             | Specimen Description| 102 | 34  | 17  | 0.857| 0.750 | 0.800|

Table 2. Micro entity and relation extraction performances. (Oracle Entities): Relation extraction results based on gold standard entities, (System Entities): Relation extraction results based on entities identified by the system.

|                | P   | R   | F1  |
|----------------|-----|-----|-----|
| System Entities| 0.903| 0.875| 0.889|
| Relations (Oracle Entities) | 0.981| 0.982| 0.981|
| Relations (System Entities)   | 0.836| 0.759| 0.795|

Table 3. Template match performances.

|                | System | Gold | Match | P    | R    | F1   |
|----------------|--------|------|-------|------|------|------|
| Templates      | 1365   | 1196 | 776   | 0.569| 0.649| 0.606|

Discussion

Error analysis revealed that rule-based entities were in general highly reliable, with f1-score performances between 0.99-1.0. Hybrid entities, likewise, had high f1-scores overall, with f1-scores ranging from 0.86-0.97. The three lowest scoring hybrid entities, were organism quantity, specimen date, no growth measurement entities, with 0.857, 0.921, 0.941 f1-scores, respectively. Organism quantity entity prediction was difficult as a result of variations in organism quantity values. For example, organism quantities were as diverse as “1100”, “1000 col/mL,” “Two colony,” “rare,” “occasional,” “no,” and “isolated.” Furthermore, if in the context of “No impinemem resistant Acinetobacteria species isolated” only “No” is highlighted as a quantity. In contrast, in the context of “Methicillin resistant Staphococcus aureus isolated”, the “isolated” is annotated. About half of false positive specimen date
entities are in passages such as “Telephoned with read back to: 010910 @1725 MONICA” while the rest appeared in report corrections. No growth measurement entities had few examples, so its performance could benefit from more training data. Meanwhile, the advice statements were where the rest of hybrid entity types committed errors.

Statistical entities had the lowest f1-scores amongst all the entities. Organism errors were partly due to appearances of entities in un-annotated parts of reports, such as report corrections, and partly due to span errors, where the simultaneous occurrence of other entity types complicated the problem. Surprisingly, UMLS semantic type features did not significantly help organism entity performance. This was attributable to discrepancies in our definition of organisms and the UMLS ontology. For example, we define “beta-lactamase negative” and “coagulase negative” in organisms whereas they are tagged “qualitative concept” by MetaMap. Though exact entity match was 0.827 f1-score, we calculated an inexact match of 0.907 f1-score. Specimen description errors occurred because of the simultaneous large variations of values and fewer examples in training. In contrast to organism entities, specimen description entities were able to leverage more generalizable features such as part-of-speech and UMLS semantic types to increase performance. Inexact match scoring resulted in a 0.863 f1-score over the 0.800 f1-score for exact.

Relation extraction was highly successful given oracle entities, but dropped dramatically with system entities, suggesting that most errors were predominantly due to entity error propagation. Thus, whenever entity text-spans were inaccurate, any relations associated with them were immediately marked wrong. Similarly, for entities that should not exist, any relation assigned to it was considered an error. The bulk of relation errors were associated with equivalenceOf relations, hasOrganismQuantity when multiple organisms or quantities were present, and hasDrugDesc, hasDrugResistance, and hasMIC relations when multiple drugs were present.

Conclusion

In this paper, we described a promising hybrid system to extract laboratory culture information from microbiology reports. One limitation was the use of only one physician to develop the schema, which leaves potential for missing information. As future work, we will continue to improve the overall template extraction performance based on our error analysis. We plan to use our microbiology extraction as part of an ongoing critical illness phenotype extraction project in our institution. Our system can also be used as an upstream process for other clinical applications as well.

Acknowledgements

We would like to thank our annotator for annotating our corpus. This project was partially funded by National Library of Medicine (T15LM007442), University of Washington Research Royalty Fund, and the National Center For Advancing Translational Sciences (UL1TR000423).

References

1. Baron EJ, Miller JM, et al. A Guide to Utilization of the Microbiology Laboratory for Diagnosis of Infectious Diseases: 2013 Recommendations by the Infectious Diseases Society of America (IDSA) and the American Society for Microbiology (ASM). Clin Infect Dis 2013;57(4):e22-e121.
2. Bellini C, Petignat C, et al. Comparison of automated strategies for surveillance of nosocomial bacteremia. Infect Control Hosp Epidemiol 2007;28(9):1030-1035.
3. Graham PL, San Gabriel P, Lutwick S, Haas J, Saiman L. Validation of a multicenter computer-based surveillance system for hospital-acquired bloodstream infections in neonatal intensive care departments. Am J Infect Control 2004;32(4):232-234.
4. Jones M, DuVall SL, et al. Identification of methicillin-resistant Staphylococcus aureus within the nation’s Veterans Affairs medical centers using natural language processing. BMC Med Inform Decis Mak. 2012;12:34.
5. Matheny ME, FitzHenry F, Speroff T, et al. Detection of Blood Culture Bacterial Contamination using Natural Language Processing. AMIA Annu Symp Proc. 2009:2009:411-415.
6. Stenetorp P, Pyysalo S, Topić G, Ohita T, Ananiadou S. BRAT: A Web-Based Tool for NLP-Assisted Text Annotation.
7. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J. LIBLINEAR: A library for large linear classification. J Mach Learn Res. 2008;9(9):1871-1874.
8. Okazaki N. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). 2007. Available at: http://www.chokkan.org/software/crfsuite/.
9. openNLP. openNLP. Available at: http://openlp.apache.org/.
10. Aronson AR. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. 2001.
11. Bejan C, Vanderwende L, Evans HL, Wurfel MM, Yetisgen-Yildiz M. On-time clinical phenotype prediction based on narrative reports. Proc Am Med Inform Assoc Fall Symp. 2013.