

Computational Algorithms that Effectively Reduce Report Defects in Surgical Pathology

Jay J. Ye, Michael R. Tan

Dahl-Chase Pathology Associates, Bangor, Maine, USA

Received: 22 March 2019

Accepted: 23 April 2019

Published: 01 July 2019

Abstract

Background: Pathology report defects refer to errors in the pathology reports, such as transcription/voice recognition errors and incorrect nondiagnostic information. Examples of the latter include incorrect gender, incorrect submitting physician, incorrect description of tissue blocks submitted, report formatting issues, and so on. Over the past 5 years, we have implemented computational algorithms to identify and correct these report defects. **Materials and Methods:** Report texts, tissue blocks submitted, and other relevant information are retrieved from the pathology information system database. Two complementary algorithms are used to identify the voice recognition errors by parsing the gross description texts to either (i) identify previously encountered error patterns or (ii) flag sentences containing previously-unused two-word sequences (bigrams). A third algorithm based on identifying conflicting information from two different sources is used to identify tissue block designation errors in the gross description; the information on actual block submission is compared with the block designation information parsed from the gross description text. **Results:** The computational algorithms identify voice recognition errors in approximately 8%–10% of the cases and block designation errors in approximately 0.5%–1% of all the cases. **Conclusions:** The algorithms described here have been effective in reducing pathology report defects. In addition to detecting voice recognition and block designation errors, these algorithms have also been used to detect other report defects, such as wrong gender, wrong provider, special stains or immunostains performed but not reported, and so on.

Keywords: Computational algorithms, error reduction in pathology, pathology report defects

INTRODUCTION

It is well known that the quality of pathology reports is crucial for patient safety and quality of patient care. Many aspects of the quality of pathology reports, such as interpretive diagnostic errors and specimen identification errors, have been addressed in the literature.^[1–8] The College of American Pathologists Pathology and Laboratory Quality Center and the Association of Directors of Anatomic and Surgical Pathology have published a guideline for reducing interpretive diagnostic error in surgical pathology and cytopathology.^[2] The College of American Pathologists Pathology and Laboratory Quality Center and the National Society for Histotechnology have published a guideline for uniform labeling of blocks and slides in surgical pathology to reduce the risk of introducing specimen identification errors during the process from specimens to slides.^[3]

In comparison to the interpretive diagnostic errors and specimen identification errors, report defects are less likely to

cause harm to patients. This type of error was also discussed in some publications.^[6,7,9] Proofreading was mentioned as an approach to reduce typographical errors.^[6,7]

We converted our dictation system from secretarial transcription to voice recognition in 2012. This conversion resulted in an increase in the number of text input errors (voice recognition errors) in the reports. During the past 5 years, we have incrementally introduced various computational approaches to identify and correct these voice recognition errors, as well as errors in certain nondiagnostic information, such as block designation errors in the gross description and inadvertent omission of immunostain or special stain results in the diagnosis or comment. These approaches have been

Address for correspondence: Dr. Jay J. Ye,
417 State Street, Suite 540, Bangor, Maine 04401, USA.
E-mail: jye@dahlchase.com

This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms.

For reprints contact: reprints@medknow.com

How to cite this article: Ye JJ, Tan MR. Computational algorithms that effectively reduce report defects in surgical pathology. *J Pathol Inform* 2019;10:20.
Available FREE in open access from: <http://www.jpathinformatics.org/text.asp?2019/10/1/20/261953>

Access this article online

Quick Response Code:



Website:
www.jpathinformatics.org

DOI:
10.4103/jpi.jpi_17_19

implemented in AutoHotkey (www.autohotkey.com, accessed March 17, 2019), Word VBA (<https://docs.microsoft.com/en-us/office/vba/api/overview/word>, accessed March 17, 2019), and R (<https://www.r-project.org>, accessed March 17, 2019). These programs have been used for error detection in both the preliminary and final reports. Although there are many types of errors the programs catch, there are only three underlying algorithms that the programs rely on.

To demonstrate how these algorithms work, we will selectively describe the computational approaches implemented in R and used by our pathologists' assistants to correct errors in the gross description and clinical information sections of the reports.

MATERIALS AND METHODS

A typical computer workstation is a desktop PC HP Elitedesk (Hewlett-Packard, Palo Alto, California) with Intel (R) (Intel, Santa Clara, CA) Core (TM) i7-670 CPU @ 2.80 GHz and 24.0 GB random-access memory. The pathology information system is PowerPath 10.0.0.19 (Sunquest Information Systems, Tucson, AZ, USA), with Advanced Material Processing (AMP module). The backend database management system for PowerPath is Microsoft SQL server.

For report preparation by pathologists' assistants or pathologists, voice recognition software is Dragon Medical Practice Edition version 11 or 12 (Nuance Communications, Burlington, MA, USA). Custom-created scripts using two programming languages, AutoHotkey (www.autohotkey.com, accessed March 17, 2019) and Dragon Advanced Scripting Language, are used to perform the dictations.

Open source programming language R version 3.5.1 (<https://www.r-project.org>, accessed January 3, 2019) is used for both interacting with PowerPath database and programming a web application for error detection. RStudio Version 1.2.1194 (<https://www.rstudio.com>, accessed March 17, 2019) is the integrated development environment used to develop the R programs.

The process of obtaining data from the pathology database using a database connectivity package (RODBC) is as described previously.^[10] Briefly, a connection string containing information on database server address, the name of the database, user login name, and password were constructed. Furnishing connection string and SQL query as two arguments to an RODBC function `sqlQuery()` retrieves the data of interest from the PowerPath database into R.

A web application, designated as "Report Checker", was developed in-house using R with shiny package (<https://shiny.rstudio.com>, accessed March 17, 2019) for the pathologists' assistants to use. Report checker is hosted on a virtual Windows Server (Windows 2012R2, 4 cores and 8 GB of RAM) in the intranet and can be accessed from any PC within the network by using a web browser. The users can query the pathology database by a date range to see if there are any cases flagged by the Report Checker for possible report defects.

The first algorithm identifies previously encountered error patterns. This algorithm is divided into two subcategories depending on how the error patterns are represented: string literals and regular expressions.

For string literal, the errors are exactly the same each time. These are some repeatable examples we have encountered: "maternal ileum" (terminal ileum), "polyps lymph node" (possible lymph node), "Native sections" (Representative sections), and so on. In addition, we use pairs of square brackets "[" and "]" with enclosed texts as place holders in our templates for voice dictation. On occasion, these place holders are left in the reports. Furthermore, some voice commands are occasionally transcribed as text, such as "Switch to Word" and "End gross."

We have noticed that the measurements of the specimen dimensions in the gross description tend to be error prone. For example, the measurements can miss the unit "cm" (dropping "cm" or "cm" being replaced by other phonetically similar words) or "x" between the dimensions transcribed as "by." Because each time, the numerical measurements and the phonetically similar words can be different, using exact text is not an efficient way to identify these errors. Instead, a regular expression is used. For the above examples, the regular expressions "x [0-9]{1,2}[.][0-9] [(a-bd-z)][(a-ln-z)][(a-zA-Z,.]{5})?" and "[0-9]{1,2}[.][0-9] by" are used to catch missing "cm" and "by", respectively.

Regardless of whether the error patterns are represented with string literals or regular expressions, the error patterns are saved in a text file, with each known error pattern occupying a single line. The R program reads these known errors into the program. The retrieved report texts are parsed for the presence of these errors. When any of these errors are encountered in the text, the results are presented to the user.

The second algorithm flags sentences containing previously unseen bigrams in the report text. Bigrams are two-word sequences within a sentence. For instance, the sentence "The specimen is bisected" contains three bigrams: "The specimen", "specimen is", and "is bisected".

Gross description texts from a 2-year period consisting of 96 thousand cases (approximately 200 thousand specimens) were retrieved from the database. These texts were preprocessed to reduce the number of unique bigrams in the texts. The conventional beginning of the gross description text "Received in formalin identified as", patients' names, dates of birth, specimen designations, and all capitalized texts were removed. For the Arabic numerals in the text, except for a single digit with an actual number of "1", all the other digits were converted to "8". This resulted in a total of slightly over 50 thousand unique bigrams. These unique bigrams were listed in a text file in the descending order of frequency, with each bigram in a single line. These bigrams were sequentially, visually inspected. When a bigram appeared to be likely from an incorrect sentence was noted, sentence(s) containing these bigrams were inspected to see if there were indeed

errors in the sentence. RStudio has the capability to display searchable data frames. This capability enabled one to type in the suspicious bigram and pull out all the sentences associated with that bigram for further inspection. This painstaking process identified slightly less than 3% of all the bigrams as bigrams associated with text input errors. Removal of these bigrams yielded a library of “normal” bigrams with consisting of slightly less than 50 thousand entries, with each entry occupying a single line in a text files.

The newly dictated gross description texts are retrieved and preprocessed in the same way as the text used for the construction of “normal” bigram library. Sentences with bigram(s) not present in the library are flagged by the program and presented to the user for inspection.

The third algorithm relies on identifying inconsistencies between the information from two different sources. Information related to tissue block submission is represented both as unstructured free text in gross description and structured data within a table in the database. When the information from these two sources conflict, the structured data always contain the correct information.

Our block designation consists of a specimen number component of Arabic numeral followed by a block number component of the capitalized alphabet. For example, the first block of the first specimen is 1A, the second block of the first specimen 1B, the 27th block of the first specimen 1AA, the third block of the second specimen 2C, and so on.

In contrast to the first two approaches for text input error detection, where parsing the text alone is sufficient, some (not all) block designation errors in gross description text cannot be detected by parsing the gross text alone. For instance, it will not be detectable if four blocks (2A-2D) are submitted for specimen 2, but the corresponding gross description says “submitted in 2A-2B.” Some errors are detectable by reading the text alone, such as designated the single block submitted for specimen 3 as “1A” in the gross description (the correct designation is “3A”).

The block information is retrieved from the PowerPath data table “acc_block”, and the information on the last block for each specimen is obtained. The text of gross description is separately retrieved. For each specimen, the program checks to see if the corresponding gross description text contains the designation of the first block, the last block, and designation for extra blocks (up to three blocks more than the last block). The program also checks to see if the block designation between the first and last block is complete and without ambiguity. For example, if the data retrieved from “acc_block” indicates that the last block for specimen 2 is 2E, the gross description text should contain both “2A” and “2E”, and should not contain “2F”, “2G”, or “2H”. Any deviation from this anticipated pattern results in the specimen being flagged and the displaying of the reason for flagging. The program only checks the specimens with 23 or fewer blocks to avoid the programmatic

complexity of dealing with block designation containing duplicated alphabets. The absolute majority of the specimens have 23 or fewer blocks submitted.

RESULTS

Our surgical pathology reports consist of three sections: Diagnosis/Comment, Clinical Information, and Gross Description. The Clinical Information (obtained from the requisition) and Gross Description sections are dictated by the pathologists’ assistants; the Diagnosis/Comment section is dictated by the pathologists.

In the gross room, the computational error checking is performed in two ways. First, at the end of gross dictation for each case, the voice command used by the pathologists’ assistants to save the dictation triggers a program implemented in AutoHotkey and Word VBA to perform error checking/correcting before the case is saved (the details are beyond the scope of this article). Second, at least once a day, the pathologists’ assistants run the Report Checker to identify additional errors that have not been caught by the aforementioned program triggered at the end of the gross dictation for each case or corrected by the pathologists’ assistants through proofreading.

Figure 1 shows a screenshot of the simplified Report Checker before one starts querying the database. After clicking the button “Start querying database”, the Report Checker will check all the reports accessioned within the specified date range (inclusive). The query results are displayed in the format of tables, with each entry containing case number and the name of the pathologists’ assistant responsible for the case, in addition to the relevant error information. This way, a pathologists’ assistant only needs to review and correct his or her own cases.

To detect the voice recognition errors in the clinical information/gross description, an algorithm to identify the known error patterns has been implemented both in a

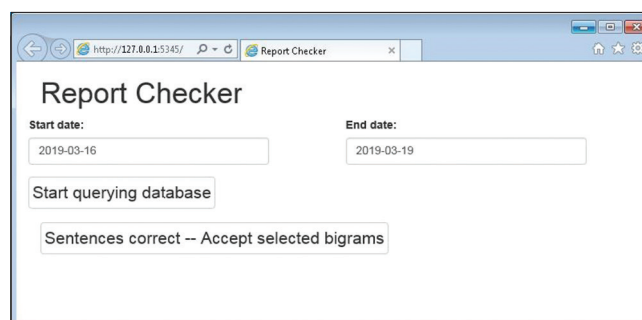


Figure 1: Screenshot of Report Checker before starting the query. The appearance of the Report Checker on the screenshot is slightly different from the actual Checker in that it is simplified and has larger font size for the demonstration purpose. The default start date and end date can be changed by the users. Clicking the button “Start querying database” will initiate the query according to the specified date range. The button “Sentences correct – Accept selected bigrams” is for the users to add the selected previously unseen bigrams to the repertoire of normal bigrams

program written in AutoHotkey and Word VBA and in the web application written in R. The exact error catching rate of this approach is thus difficult to quantify, but conservatively estimated by the pathologists' assistants to be 4%–8%. The examples of errors caught include: “BCC” typed as “BBC”, “Path pending” typed as “Passed pending”, measurement unit “cm” typed as “sodium”, missing measurement unit “cm”, “perpendicular sections” typed as “radicular sections”, “chin” typed as “shin”, and so on. Table 1 shows additional actual examples identified by the known error patterns. The corresponding correct sentences are also displayed.

The bigram approach is implemented in the web application Report Checker only, that is, not in the command before saving the gross description for each case; therefore, its error detection rate can be more precisely estimated. To this end, gross description texts (1717 cases) dictated during a 2-week period before the bigram algorithm had been implemented and after the 2-year period from which texts were used for the construction of normal bigrams were examined. In these texts, the algorithm identified 0.5% of bigrams that were not in the “normal” bigram library, flagging 3% of the sentences in the gross description. Ten percent of the flagged sentences contained errors, corresponding to 1.9% of the cases. Half of these errors could be confusing while the other half were minor. The confusing examples include “where from margin” (away from margin), “uterine process” (uterine corpus), “actually covered” (partially covered), “port material” (soft material), and so on. The very minor ones include “is an blue” (is a blue), “are 2 portion of” (are 2 portions of), “is it is 0.3 × 0.2 cm” (is a 0.3 × 0.2 cm), and so on.

After being incorporated into the Report Checker, the bigram algorithm has identified many errors on the ongoing basis, such as “Than mature cavity is focally scarred” (Unseen bigram: “Than mature”, Correct sentence: “The endometrial cavity is focally scarred”) and “A well-defined mass identified” (Unseen bigram: “mass identified”, Correct sentence: “A well-defined mass is not identified”). Additional examples are shown in Table 2.

For the block designation errors, it is difficult to precisely estimate the prevalence corrected by the algorithm since these errors are identified using both an Autohotkey voice command and the web application Report Checker. It is conservatively estimated using historical data and by the pathologists' assistants that 0.5%–1% of all the cases contain a block designation error that is detected by the combined approach. The error could be denoting a block with a wrong specimen number, such as designating “1A” for specimen 5, which should have been “5A.” Within a single specimen, the type of errors includes describing fewer blocks than actual submissions, more blocks than actual submissions, or ambiguity in the designation.

Table 3 is a deidentified rendition of block designation error output table with an added interpretation (“Actual findings”) column. Specimen number, last block, first block, last block,

Table 1: Examples of errors detected by known error patterns

Known error pattern	Portion of sentence with error	Portion of sentence corrected
×0.1 to inferior	1.2×0.1 to inferior and green,	1.2×0.1cm. Inked green,
tan measuring	2 soft tan measuring	2 soft tan cores measuring
crested nodule	tan-brown, crested nodule	Tan-brown, crusted nodule
white white	gray-white white, rubbery fibrous	gray-white, rubbery fibrous
×0.9 tan ann	2.5×2.0×0.9 tan annular	2.5×2.0×0.9 cm annular
viscus bile	dark green viscus bile	dark green viscous bile
cm cm	0.4×0.4×0.3 cm cm red-tan	0.4×0.4×0.3 cm red-tan
fimbria addendum	the entire fimbria addendum	the entire fimbriated end

Table 2: Examples of errors detected by the bigram algorithm

Unseen bigram	Portion of sentence with error	Portion of sentence corrected
white male	grey white male	grey white nail
pale entirely	pale entirely serially submitted	nail entirely serially submitted
m in	0.6 m in diameter	0.6 cm in diameter
car sequentially	a car sequentially between	in cassettes sequentially between
integrated pink	centrally integrated pink skin	centrally indurated pink skin
of with	wedge of with a 6.5 cm linear	wedge of lung with a 6.5 cm linear
an spongy	the parenchyma is tan an spongy	the parenchyma is tan and spongy
whie tan	whie-tan cylindrical fragments	white-tan cylindrical fragments

extra block, and internal block checking are the column names of the block checking output table in the Report Checker. The output table also contains the accession number of the case, which is omitted here for deidentification purposes. The column “Actual findings” in Table 3 is not a part of the output table; it is the interpretation of the output.

DISCUSSION

Our group has established many policies and practices as well as utilizing barcoding technology to ensure the quality of our practice, reflecting our emphasis on reducing the risk of interpretive errors, and maintaining specimen identification throughout the process. Many of our practices are similar to the ones proposed in the guidelines.^[2,3]

Table 3: Deidentified rendition of block designation error output table with an added column of actual findings

Case	Specimen number	Last block	First block	Last block	Extra block	Internal block checking	Actual findings
1	6	6C	Yes	Yes	Yes	Ok	Nonexisting block "6D" mentioned in gross
2	10	10A	No	No	No	Ok	Gross stated "10 A" (with a space between 10 and A), not a real error
3	1	1B	Yes	No	Yes	Ok	Actual blocks submitted "1A" and "1B", but gross stated "1A-1C"
4	2	2A	No	No	No	Ok	Block "2A" was designated as "1A" in gross
5	1	1H	Yes	No	No	Ok	No space between 1H and the ensuing text
6	1	1B	Yes	No	No	Ok	Actual blocks submitted "1A" and "1B", but gross only mentioned "1A"
7	4	4C	No	No	No	Ok	Blocks for specimen #4 was designated as "5A-5B" and "5C", so that both specimens 4 and 5 were flagged
7	5	5B	Yes	Yes	Yes	Ok	See row above
8	2	2A	Yes	No	No	Ok	False alarm, no error, not sure why last block query result is "No"
9	1	1M	Yes	Yes	No	Ambiguous 1C	Gross has ambiguous "1B-1C" and "1C-1D", with meaning of 1C being ambiguous
10	1	1S	Yes	Yes	No	Missing 1E	"1E" not designated in gross

The focus of this article is on report defects. Despite the attentions and efforts, a small percentage of reports still unavoidably contain report defects, particularly due to voice recognition errors. Over the past 5 years, we have incrementally expanded the scope of using a computational approach to detect these defects.

Three algorithms presented here underlie the computational approaches to detect other types of errors too. For instance, identifying conflicting information approach has also been used to detect the following errors: wrong gender assignment in the pathology information system, wrong provider with similar appearing names being entered into the system at accessioning, final reports without mentioning immunostains or special stains that have been performed, and so on. Identifying known patterns of errors has been used by pathologists for the final diagnosis text error detection; it was previously described without going into the underlying algorithm.^[11]

With the algorithm identifying the known error patterns, the ability of the programs to detect errors increases over time, as users put more and more known error patterns into the text file that the programs rely on. This approach only catches errors that the users have entered into the text file. Nevertheless, whenever the program identifies something as an error, it is highly specific.

The bigram approach is paradigmatically opposite to the algorithm identifying known error patterns. While the error pattern identifying algorithm relies on a defined list of errors, the bigram approach relies on the definition of "normal" bigrams, which is a list of 50 thousand bigrams that were used during a 2-year period. Because the bigram algorithm relies on the knowledge of what is normal and does not rely on what abnormal patterns are, it can catch errors that occur for the very first time without needing to know exactly what they are. It makes sense that these two algorithms are complementary in detecting typographical/voice recognition errors. The inevitable

downside for the bigram approach is that the specificity is low, approximately 10%. Fortunately, the pathologists' assistants do not find this method burdensome; the web application was designed in such a way that the pathologists' assistants only need to deal with the false alarm once by selecting these entries and then clicking a button on the browser [the lowest button in Figure 1] for the application to learn the entries as normal bigrams. In addition, only 3% of the sentences are flagged and the bigrams point to the relevant portions of the dictation.

A trigram approach was also tested; it required checking twice as many sentences as the bigram approach without perceptible increase in the error detecting capability (data not shown).

Certain report defects, such as some block designation errors as well as immunostains performed and interpreted but not reported, are not discoverable by reading the report text alone. Information from other sources, such as the actual submission of the tissue blocks or the actual immunostains performed are required to know if the report text is incorrect or incomplete. This is the underlying rationale for error checking through identifying inconsistency between information obtained from two sources.

The prevalence of report defects probably tends to be underestimated; at least it is true for us. For instance, the bigram approach is the latest addition to our error checking program after all other approaches have been introduced and after the error pattern text files have accumulated many entries over the years. We were surprised that the bigram approach identified additional voice recognition errors in the gross descriptions in 1.9% of the cases.

Although the aggregate prevalence of report defects is significant, the prevalence of each exact report defect is low; therefore, they can be difficult to identify, similar to a needle in a hay stack situation. Computers can go over a large quantity of data in a short span of time in accordance with any predefined

rules unerringly. The effectiveness and the efficiency of the computational approaches are thus not surprising.

Our experience shows that the use of computational approaches to detect report defects is effective and costs very little additional time. These approaches have identified many report defects that would have otherwise evaded human detection by proofreading.

CONCLUSIONS

Identifying known error patterns and flagging sentences containing previously unseen bigrams are two complimentary algorithms effective in detecting typographical/voice recognition errors in the report texts. Identifying conflicting information is an effective algorithm that can be used to detect many types of nondiagnostic information in or associated with the reports.

The intent of the article is to share the general principles of the computational approaches that we have stumbled upon and found effective. Their implementations in other pathology practices/departments do not necessarily have to be in the language we use. Any languages that the local expertise possesses can potentially be used to implement these algorithms.

Financial support and sponsorship

Nil.

Conflicts of interest

There are no conflicts of interest.

REFERENCES

1. Laposata M, Cohen MB. It's our turn: Implications for pathology from the institute of medicine's report on diagnostic error. *Arch Pathol Lab Med* 2016;140:505-7.
2. Nakhleh RE, Nosé V, Colasacco C, Fatheree LA, Lillemoe TJ, McCrory DC, *et al.* Interpretive diagnostic error reduction in surgical pathology and cytology: Guideline from the college of American Pathologists Pathology and Laboratory Quality Center and the Association of Directors of Anatomic and Surgical Pathology. *Arch Pathol Lab Med* 2016;140:29-40.
3. Brown RW, Della Speranza V, Alvarez JO, Eisen RN, Frishberg DP, Rosai J, *et al.* Uniform labeling of blocks and slides in surgical pathology: Guideline from the college of American Pathologists Pathology and Laboratory Quality Center and the National Society for Histotechnology. *Arch Pathol Lab Med* 2015;139:1515-24.
4. Nakhleh RE. Patient safety and error reduction in surgical pathology. *Arch Pathol Lab Med* 2008;132:181-5.
5. Renshaw AA, Gould EW. Measuring errors in surgical pathology in real-life practice: Defining what does and does not matter. *Am J Clin Pathol* 2007;127:144-52.
6. Sirota RL. Defining error in anatomic pathology. *Arch Pathol Lab Med* 2006;130:604-6.
7. Nakhleh RE. Error reduction in surgical pathology. *Arch Pathol Lab Med* 2006;130:630-2.
8. Layfield LJ, Anderson GM. Specimen labeling errors in surgical pathology: An 18-month experience. *Am J Clin Pathol* 2010;134:466-70.
9. Zarbo RJ, Meier FA, Raab SS. Error detection in anatomic pathology. *Arch Pathol Lab Med* 2005;129:1237-45.
10. Ye JJ. Pathology report data extraction from relational database using R, with extraction from reports on melanoma of skin as an example. *J Pathol Inform* 2016;7:44.
11. Ye JJ. Artificial intelligence for pathologists is not near – it is here: Description of a prototype that can transform how we practice pathology tomorrow. *Arch Pathol Lab Med* 2015;139:929-35.