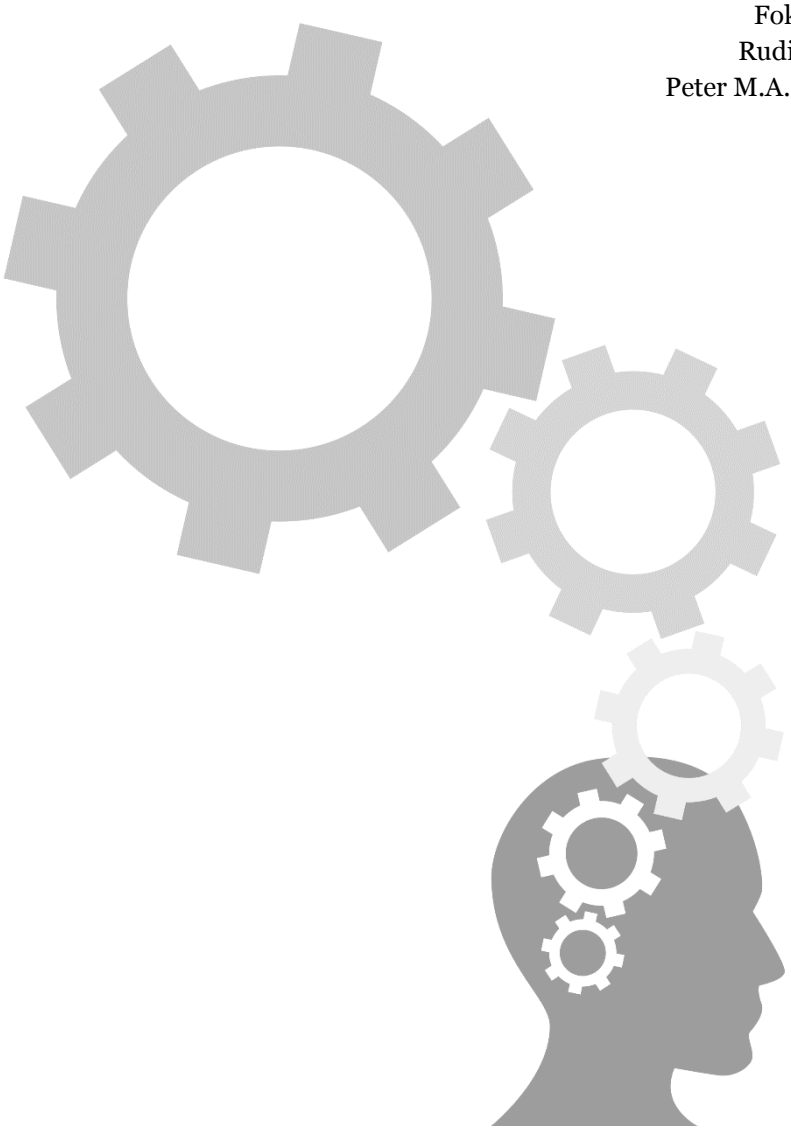
8

Automatically structuring free-text radiology reports using a machine learning algorithm

Joost Timmerman
Wiard Jorritsma
Fokie Cnossen
Rudi A. Dierckx
Peter M.A. van Ooijen



Abstract

Objectives: To develop a system that automatically converts dictated free-text radiology reports into structured, standardized reports. Such a system would be an interesting alternative to conventional structured reporting systems, because it requires no additional actions from the radiologist and does not interfere with image viewing and interpretation.

Methods: Two report templates were developed based on literature and interviews with clinicians. 165 free-text radiology reports on the malignant lymphoma were annotated. A computational system using a Linear Chain Conditional Random Fields (LC-CRF) machine learner, specifically designed for learning sequences, was trained on classifying information in the annotated free-text reports. A post-processing step was added to correct specific tokens that were misclassified by the machine learner. The classified texts in the free-text reports were automatically re-structured into the templates to form standardized, structured reports. The system's classification performance was assessed by calculating the average F-score over five combinations of training and test sets.

Results: The system was able to correctly classify most information in the free-text reports. It obtained F-scores of 88.18 (micro-averaged) / 87.85 (macro-averaged) on correctly classifying texts. The post-processing step improved performance to F-scores of 89.30 (micro averaged) / 88.60 (macro averaged).

Conclusions: A machine learning system based on the LC-CRF algorithm can be used to automatically structure and standardize the information contained in free-text radiology reports. However, more research is needed to improve the performance of the system to a level that is acceptable in clinical practice, apply it to different classes of reports, and extract more detailed information from the reports.

Mastery of language affords one remarkable opportunities.

— Alexandre Dumas

Introduction

In radiological reports, diagnostic questions are answered by a radiologist by providing an overview of findings, impressions and diagnosis of acquired radiological images. These reports are normally dictated and constructed during the interpretation of the images. However, there are several problems with the current situation of free-text radiology reporting.

First, the information in the free-text reports is 'locked in'. Although the information is present in the reports, the report database cannot be searched effectively. This may introduce significant workflow bottlenecks or even barriers. For example, answering the question 'How did tumor A of patient X develop in the past year?' is only possible by manually going through all reports of patient X of the past year searching for measurements of the tumor in question, and then calculating the development by hand.

A second problem is that free-text reporting leads to a high variability in reports. Clinicians therefore have to adapt to the way each report structures and presents information. This could cause critical information to be overlooked or misinterpreted, which can result in severe medical errors.

A solution to these problems is to use structured reporting systems that let radiologists report in a controlled fashion. For example by having them fill in the blanks in a standard report template (well-known report templates are those made available by the Radiology Reporting Initiative of the Radiological Society of North America [1]), and/or having them use a standardized lexicon (e.g. RadLex [2] or BI-RADS [3]).

However, studies comparing structured reporting to conventional free-text reporting show mixed results [4-9]. A major argument against reporting in a structured fashion is that it would require additional actions by the radiologist, which cost time and might interfere with the interpretation of the images [5]. However, other studies show a significant increase in content satisfaction and clarity satisfaction when comparing structured reporting to conventional reporting [7].

An alternative approach is to develop a system that automatically converts free-text reports into a structured format. The global outline of a possible system is shown in Fig. 1. The system would receive a free-text report as input, classify the text in the report into blocks belonging to different categories using natural language processing techniques, and use these blocks to compose a structured version of the report. This approach has the advantage that it yields structured reports without changing the way radiologists report their findings, allowing them to devote their full attention to their primary task: viewing and interpreting images.

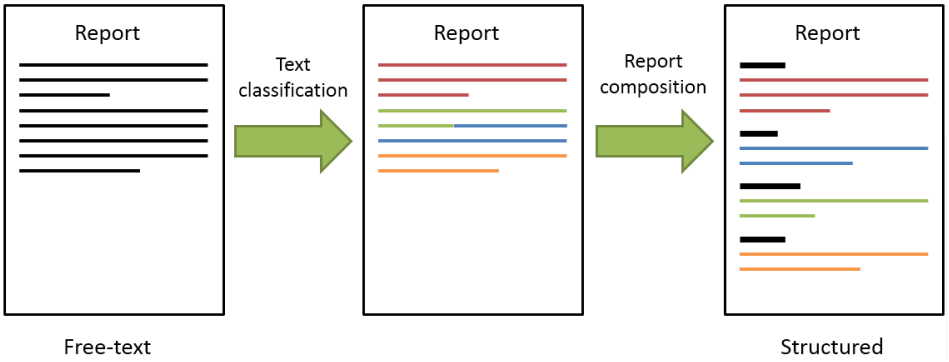


Figure 1. Global outline of a system that automatically converts free-text radiology reports into a structured format. The system receives a free-text report as input, classifies the text in the report into blocks belonging to different categories using natural language processing techniques, and uses these blocks to compose a structured version of the report.

The key to the system's success is its ability to correctly classify the text in the free-text report. Once the text is classified, converting the report into a structured format is a trivial task. Studies using natural language processing techniques to extract information from free-text radiology reports have shown promising results [10], which suggests that it is possible to develop an automatic report-structuring system.

In this study we aimed to develop a system that converts free-text radiology reports into a structured format using a machine learning algorithm and to evaluate its text classification performance.

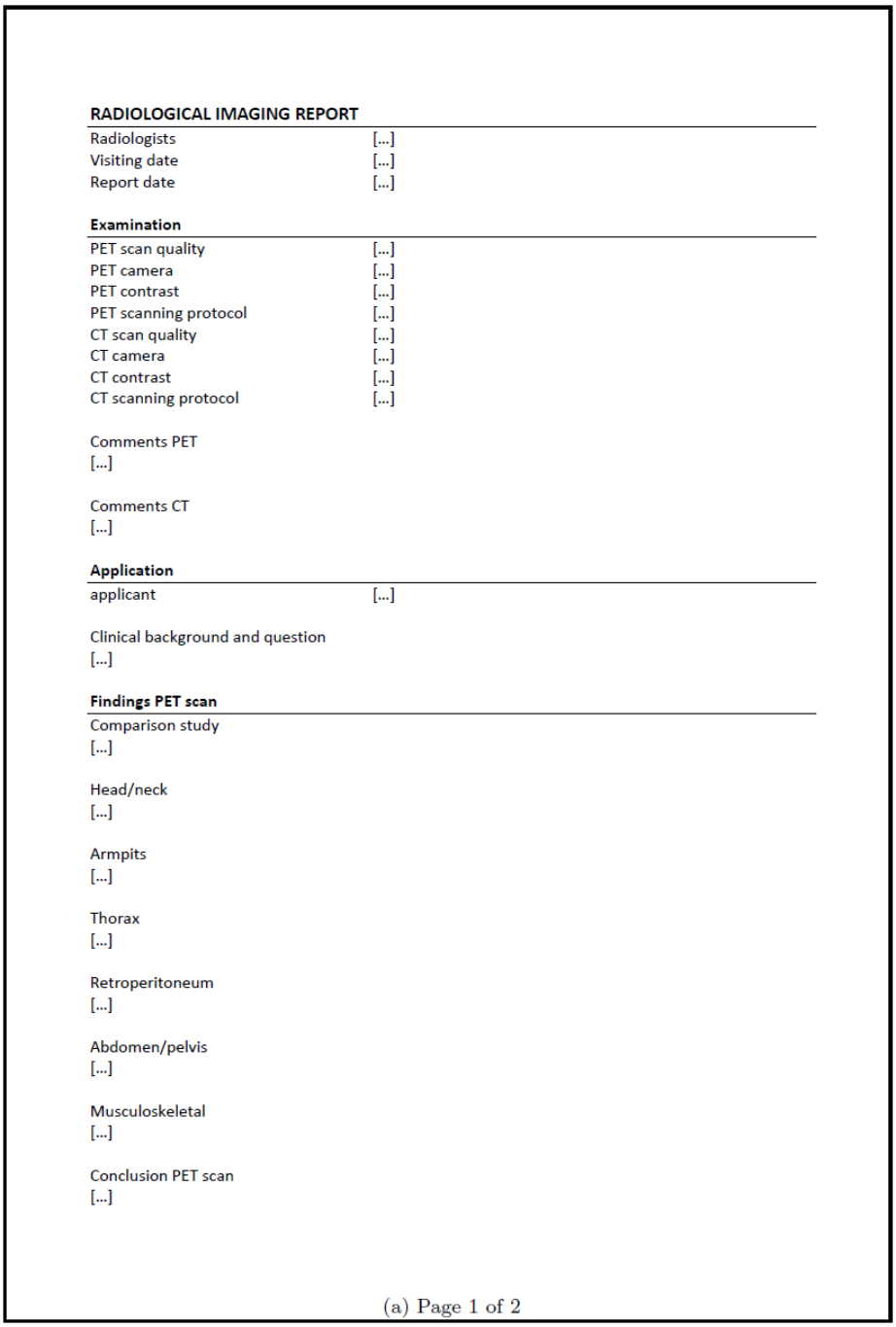
Methods

Text classes

For the purpose of this study we only used radiology reports on the malignant lymphoma. First, the ideal structure and required content of a report on the malignant lymphoma were determined in interviews with referring clinicians. This resulted in two templates representing the two most preferred radiology report structures (one is shown in Fig. 2). While the overall content is exactly the same, the difference between the two global structures is the position of the conclusion section: in the first structure, the conclusion section is positioned after the findings, while in the second structure the conclusion section precedes the findings.

Fifteen distinct text classes were identified based on the information in these templates (Abdomen/Pelvis, Camera, Conclusion, Contrast, Head/Neck, Musculoskeletal, Armpits, Person (executing radiologist), Retroperitoneum, Scan

quality, Scanning methods (comments), Scanning protocol, Thorax, Comparison, and Clinical Background and Question).



RADIOLOGICAL IMAGING REPORT

Radiologists	[...]
Visiting date	[...]
Report date	[...]

Examination

PET scan quality	[...]
PET camera	[...]
PET contrast	[...]
PET scanning protocol	[...]
CT scan quality	[...]
CT camera	[...]
CT contrast	[...]
CT scanning protocol	[...]

Comments PET
[...]

Comments CT
[...]

Application

applicant	[...]

Clinical background and question
[...]

Findings PET scan

Comparison study
[...]

Head/neck
[...]

Armpits
[...]

Thorax
[...]

Retroperitoneum
[...]

Abdomen/pelvis
[...]

Musculoskeletal
[...]

Conclusion PET scan
[...]

(a) Page 1 of 2

Findings CT scan

Comparison study
[...]

Head/neck
[...]

Armpits
[...]

Thorax
[...]

Retroperitoneum
[...]

Abdomen/pelvis
[...]

Musculoskeletal
[...]

Conclusion CT scan
[...]

Integrated conclusion

[...]

(b) Page 2 of 2

Figure 2. Report template of the most preferred radiology report structure based on outcomes of interviews with referring clinicians.

Machine learner

After the text classes were determined, we needed an algorithm that could identify them in a free-text report. For this we used a machine learning algorithm called Linear Chain Conditional Random Fields (LC-CRFs) [11]. CRFs form a modelling method specifically used for structured prediction. Ordinary classifiers such as Naive Bayes base their prediction on a single instance without taking features of the neighboring instances into account, but CRFs can take context into account. The Linear Chain CRF that is popular in natural language processing can predict sequences of labels for sequences of input instances. Within this study, these sequences of input are formed by words and punctuation that together form sentences.

For the machine learning implementation, the open-source CRF++ software (version 0.58) [12] was used.

Training and test sets

To construct a dataset, Dutch free-text radiology reports on the malignant lymphoma were collected in our hospital. A total of 165 of these reports were annotated by a single annotator for the fifteen classes using GATE (General Architecture for Text Engineering) [13][14]. The annotated reports were exported from GATE as Extensible Markup Language (XML) files with the annotation labels in inline format.

To form the training and test sets, the inline annotations were converted to the input file format of the CRFs machine learner using a small piece of JAVA code. The input file format of the CRF machine learner required words and punctuation signs to be on separate rows. To identify sentence boundaries, an empty line was put between two rows. Each token could furthermore be followed by zero or multiple features. The last column of the row held the label associated with the row's token. The CRF machine learner further required every row to have the exact same amount of columns. Therefore, items in the annotated text that had no assigned label after annotation (e.g. headings, addresses, etc.) were labelled *noTag*. Therefore, the final datasets contained not 15, but 16 labels for the CRFs learner to train and test on.

To determine how accurate the CRF's predictions are in practice, a model validation technique called cross-validation was applied. Cross-validation is used to assess how the results of a predictive model will generalize to an independent dataset. In practice, this implies that separate datasets are used for training and testing purposes. For this study, the datasets were created semi-randomly to ensure an 80/20 split of data files, meaning that the training set contained approximately 80% of the data, and the test set the remaining 20% of the data. Furthermore, the

data of a radiology report could only occur in either the test set or in the training set, not in both. To avoid incidental higher scores on a specific distribution of reports in the datasets, K-fold cross-validation was applied by randomly partitioning the full dataset into k equal sized partitions. Of the k partitions, a single partition is used as the test set (validation) while the remaining $k - 1$ is used for the training set. This process is then repeated k times, in this study with $k = 5$ thus resulting in five combinations of training and test sets. The CRF is trained and tested on each of the five combinations, and results are then averaged to obtain a single estimation. The advantage of using K-fold cross-validation is that each observation is used for both training and testing, and each observation is used for testing exactly once.

Features

Multiple columns of features were added to the training and test set, the first being the identification numbers of the reports that were concatenated while forming the training and test sets. By adding these numbers, the merged radiology reports could be kept apart in the merged data files and thus be retrieved individually, or split when needed. This feature row was not used during training or testing, but it was introduced for this purpose only.

A second feature column that was added to each token held a positional feature of that token in relation to the report it was in. The hypothesis behind this is that specific words or combinations of words occur in more or less specific locations in the report. Thus, providing the CRF with the positional information about the current token is likely to improve performance. For example, the conclusion of the radiologists' findings is likely to be found in the final part of the free-text report, therefore providing the CRF with information that represents 'the current token is located in the final quarter of the report' can help to determine the correct label for that token. For the positional features, the total amount of tokens n of each radiology reports were determined and used to divide the report into x parts, with each part consisting of $n=x$ tokens. The positional feature would then be 0 if it fell in the first part, 1 if it fell in the second part, . . . , or $n-1$ if it fell in the n^{th} part.

The third feature holds the value true if the current token starts with a capital or false if this is not the case. The fourth feature holds the value true if the next token is a colon, or false otherwise. They were added based on the idea that a combination of a capital and a colon is likely to indicate a transition to another chunk of information like for example a subheading indicates.

The fifth to last feature columns (without including the annotated label column) hold characteristics about token frequencies in the labelled categories of the report. If a token is frequent in texts under a certain annotated label, it is more likely that the token needs to be assigned that label than a label under which it is not frequent

at all. First, the frequencies of tokens under each specific label were determined for the data in the entire training set.

After determining token frequencies under each label, non-unique tokens (e.g. tokens that occurred under two or more labels) were identified and iterated over. In each step only the entry of the non-unique token with the highest frequency count is retained. Entries under other labels with the same token but with a lower frequency count were removed. If two entries that shared identical tokens had an equal frequency count, both entries were retained. The idea behind removing the non-unique entries is that although a token can occur in texts with different labels, it is likely that the token is more strongly related to the label under which the word frequency is highest.

Based on the frequency data, the values of the token frequency feature columns were determined. Each label has a feature column of its own. The value in this column holds true if the current token y is an element of the Y most frequent tokens under that label. Since non-unique entries were modified in the previous step, only one of the feature columns can hold the value true for a specific token. These frequency features were added to improve results for classes with fairly specific information that scored below average in the intermediate results.

Feature templates

Since the CRF++ software is designed as a general purpose tool, it needs a feature template file. In this file, the (combinations of) features and tokens that are used in training and testing are described.

Part of a simple feature template is shown below. Each line specifies one template. In each template the macro %x[row; col] is used to specify a token in the input data. Row specifies the relative position from the current focusing token and col specifies the absolute position of the column. For unigram templates, the preposition U is used, while for bigram templates the preposition B is used. Feature templates can become as complex as is needed, so that the CRF can be trained on combinations of multiple features and tokens.

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]            < previous token
U02:%x[0,0]             < current token
U03:%x[1,0]             < next token
U04:%x[2,0]
U05:%x[0,0]/%x[0,1]     < current token / first feature
U06:%x[0,0]/%x[1,2]     < current token / second feature of next token
...
```

During training, the CRF++ software expands the templates and generates a set of feature functions. These describe which combinations of tokens (and features, if specified) have positive relations. The algorithm then learns the associations (i.e. weights) between feature attributes and labels. It does this by searching through the variable space to find local maxima and minima. The total number of functions generated by a template is equal to L*N (or L*L*N for bigram features), where L is the number of output classes and N the number of unique strings expanded from the given template. The CRF++ software outputs a model file that can later be used for testing.

The CRF testing procedure takes the model file together with the test-data as input. For the CRF to work, the test data file needs to be in the exact same format as the training file with equal amounts of columns and similar features. The CRF then outputs the information in the test file with a new column appended to the end that contains the tag or label that the CRF has assigned to the tokens. One can then evaluate the results by computing the difference between the estimated labels in the n^{th} column, and the true answer label in the $n\text{-}1^{th}$ column that was assigned during annotating.

Output enhancements

The CRFs machine learner had difficulties labelling the sentences present in the texts about the *Clinical background and question* and the *Conclusion* of the reports. This was to be expected, since these texts contain sequences of tokens and transitions between these tokens that are also likely to appear in other parts of the report. Since both the *Clinical background and question* and the *Conclusion* were found in consistent parts of the free-text reports, and were succeeded by more or less predictable parts of text (e.g. a heading or a person's name), a piece of JAVA code was developed to enhance the results in these parts of the CRF's output. When the JAVA code was ran, it automatically processed the CRF's output and outputted a new file with the resulting enhancements in an appended column.

Table 1 shows a simplified example in which the CRF has labelled part of the sentence *'This shows how the output can be enhanced!'* with the label body and part of the sentence with the label prologue. Let's think of the part that is labelled with prologue as being wrongly classified. If we write a piece of code that is triggered by the combination of the token *This* AND the CRF's assigned label *body*. After triggering the enhancer, it will output the label body and will continue to do so, thereby disregarding the CRF's initial output of the prologue label. Another trigger deactivates the enhancer. This means that from that point forwards, the CRF output is no longer overwritten. The final output file will have a similar

Table 1. Example of enhancement of the CRF output.

TOKENS	FEATURE COLUMNS	CRF LABEL	ENHANCED LABEL	
This	...	intro	intro	
is	...	intro	intro	
an	...	intro	intro	
example	...	intro	intro	
sentence	...	intro	intro	
.	...	intro	intro	
Goal	...	heading	heading	
:	...	heading	heading	
This	...	body	body	<< activation trigger
shows	...	body	body	< enhanced section
how	...	body	body	< enhanced section
the	...	prologue	body	< enhanced section
output	...	prologue	body	< enhanced section
can	...	prologue	body	< enhanced section
be	...	prologue	body	< enhanced section
enhanced	...	prologue	body	< enhanced section
!	...	prologue	body	< enhanced section
Final	...	ending	ending	<< deactivation trigger
section	...	ending	ending	
of	...	ending	ending	
the	...	ending	ending	
example	...	ending	ending	
.	...	ending	ending	
...				

formatting as the example below, with an extra column appended to the file that contains the enhanced label.

Composing the structured reports

Structured reports can be composed by ordering labelled sequences and structural elements such as (sub-)headings, page breaks and white space in any desired fashion. To form sentences from the CRFs' output after it was split, another piece of JAVA code was written. For each report that was part of the CRFs' output, this code joined successive tokens with identical labels together, while putting space characters in between. It also removed headings that were labelled by the CRF but that were no longer of any value (e.g. texts that became redundant due to the fact that template files contained new (sub-)headings). Furthermore, incorrect white space that was inside or around measurements, dates and punctuation was removed or modified to make the overall sentences more readable.

Sequences of tokens were stored under the label the CRF had assigned to the sequence. The order of CRFs' output (which was equal to the order of input) was fully maintained to avoid unintended sentence or paragraph transitions which could result in misunderstandings of the report contents. No texts were replaced in order to preserve the radiologists' intended meaning. The blocks of text assigned a

specific label could then be individually retrieved. These text blocks could then be used to compose a structured report in any format desired (e.g. using XML tags, HTML documents, or plain text files).

In this study, the CRFs' output was written to the structured templates discussed earlier. For each report two formats were composed: one with the conclusion before the section communicating the findings and one with the conclusion at the end of the report, after the findings. Finally, an iteration was performed over the entire structured report to identify whether the report contained texts on positron emission tomography (PET) and/or computed tomography (CT) information, and empty, irrelevant sections were removed. Thus, if the report did not contain any information on a PET scan at all, then all PET related sections were removed from the report to clean up the final, finished product.

Data analysis

The CRF's output was used to calculate the true positives, false positives, and false negatives (see contingency Table 2) from the differences between the CRF's assigned labels in the n^{th} column in the output, and the true answer label in the $n\text{-}1^{th}$ column of the output.

The system's overall classification performance was then determined by calculating micro- and macro-averaged F-scores (range 0-100). The F-score considers both the precision (i.e. the probability that the class has been predicted) and recall (i.e. the model's ability to select instances of the corresponding class; commonly called sensitivity) and can be interpreted as the weighted average of both. Micro-averaged F-scores are calculated by first summing up individual true positives, false positives, and false negatives of the output, and using these scores for further statistics of the F-score. Since the classification decision on each individual token counts as one, labels with higher token count will be weighted heavier in the final score (i.e. topics count proportionally to their frequency). In macro-averaged scores, equal weight is given to each class or label. The

Table 2. Contingency table specifying the possible outcomes of results versus the given 'gold standard' of the annotations in the test set.

		Condition (or 'gold standard')	
		Label X	Label other than X
Test outcome	Predicted label X	True positive	False positive (type I error)
	Predicted label other than X	False negative (type II error)	True negative

effectiveness on the large classes in the test collection is therefore better represented by the micro-averaged scores, while the effectiveness of smaller classes is better represented by the macro-averaged scores [15].

The F-scores are the average results of five combinations of training and test sets, which is the result of applying the K-fold cross validation technique to the dataset. The trained feature column indicates the training condition that was used for the CRF by specifying a specific feature template. The *none, feature only* condition therefore corresponds to using a feature template in which only the current token and neighboring tokens were taken into account during training, without specifying any extra features to train on. The *pos 2* condition refers to training the CRF with a feature template in which a positional feature that divides the report into x = 2 parts is added. The corresponding *pos 4* refers to training with a positional feature that divides the report into x = 4 parts. Note that the *none, feature only* condition is comparable to training with a positional feature that divides the report into x=0 parts, thus *pos 0*. The *pos 4 + colon + capital* condition is an extension of the *pos 4* condition by also specifying macro templates for the colon and capital features. Finally, the most extensive conditions trained were the conditions in which the word frequency features were also specified in the feature template. They are indicated by wordfreq: the *wordfreq 25*, *wordfreq 50* and *wordfreq all* conditions indicate that only the top 25, top 50 or all of the frequency counts were used when creating the feature columns.

Results

Table 3 shows the initial micro- and macro-averaged F-scores obtained by the system.

The assigned labels for the texts on the *Clinical background and question* and the *Conclusion* were enhanced by applying the post-processing algorithm discussed earlier. Post-processing the initial CRFs' results improved results on both precision and recall of most classes. The new F-scores are shown in Table 4. Since the results were drastically improved, no extended analysis will be provided on the initial CRFs' results, as they were superseded by the newer, post-processing results.

Further analysis of results on the data after the post-processing step showed overall good precision and recall on class labels (see Table 5). In the tokens only condition, the lowest scoring class labels are *Armpits* and *Retroperitoneum*. The CRF obtained a precision of 88.77% and recall of 64.19% on the label *Armpits*, and a precision of 75.37% and recall of 80.74% on the label *Retroperitoneum*. Highest scoring classes are the *Person* class, with which all personal names were annotated, and the *Camera* class, which holds the information such as brand and type of the medical imaging camera used in the examination. Furthermore, the results show

Table 3. Average F-scores calculated from the CRF output over five combinations of training and test sets. For the micro-averaged F-score, each classification decision is counted separately, while for macro-averaged F-score, equal weight is given to each class label.

	CRF output	
Trained features	F-score (micro)	F-score (macro)
None, tokens only	86.79	87.08
Pos 2	87.24	87.11
Pos 4	87.70	87.45
Pos 4 + colon + capital	**88.18**	**87.85**
Pos 4 + colon + capital +wordfreq top 25	87.24	86.99
Pos 4 + colon + capital +wordfreq top 50	87.37	87.08
Pos 4 + colon + capital +wordfreq all	87.98	87.66

Table 4. Average F-scores calculated from the enhanced CRF output over five combinations of training and test sets. For the micro-averaged F-score each classification decision is counted separately, while for macro-averaged F-score equal weight is given to each class label.

	CRF output	
Trained features	F-score (micro)	F-score (macro)
None, tokens only	89.03	88.66
Pos 2	88.35	87.81
Pos 4	88.68	88.11
Pos 4 + colon + capital	**89.30**	**88.60**
Pos 4 + colon + capital +wordfreq top 25	88.37	87.76
Pos 4 + colon + capital +wordfreq top 50	88.43	87.79
Pos 4 + colon + capital +wordfreq all	89.08	88.40

Table 5. Average precision and recall on the post-processed results from the tokens only condition of the CRF. Averages are calculated from five combinations of training and test sets.

Class label	**Precision**	**Recall**
Abdomen/pelvis	84.97 %	82.28 %
Camera	97.98 %	99.67 %
Conclusion	82.59 %	98.19 %
Contrast	93.89 %	85.74 %
Head/neck	87.29 %	82.57 %
Musculoskeletal	91.10 %	75.54 %
Armpits	88.77 %	64.19 %
Person (executing radiologist)	97.55 %	98.81 %
Retroperitoneum	75.37 %	80.74 %
Scan quality	94.30 %	79.82 %
Scanning method (comments)	96.81 %	88.52 %
Scanning protocol	93.17 %	87.06 %
Thorax	82.97 %	82.24 %
Comparison	92.37 %	85.32 %
Clinical background and question	93.54 %	96.87 %
noTag	99.24 %	99.46 %

high precision and recall for the *noTag* class label. Scores in the other conditions were comparable, with highest and lowest scoring classes identical to the aforementioned classes.

Further analysis of results on the data after the post-processing step showed that in the tokens only condition, the lowest scoring class labels are *Armpits* and *Retroperitoneum*. The CRF obtained a precision of 88.77% and recall of 64.19% on the label *Armpits*, and a precision of 75.37% and recall of 80.74% on the label *Retroperitoneum*. Highest scoring classes are the *Person* class, with which all personal names were annotated, and the *Camera* class, which holds the information such as brand and type of the medical imaging camera used in the examination. Furthermore, the results show high precision and recall for the *noTag* class label. Scores in the other conditions were comparable, with highest and lowest scoring classes identical to the aforementioned classes.

Discussion

In this study we aimed to develop a system that converts free-text radiology reports into a structured format using a machine learning algorithm and to evaluate its text classification performance. The machine learning system in place used an approach called Linear Chain Conditional Random Fields (LC-CRFs), a technique aimed specifically at sequence learning.

Results showed that the machine learning system yielded good results when trained on the sequences of words, sentences and sections in dictated free-text radiology reports. The system was trained with relatively simple features that were directly based on the input data, without the need for expert knowledge. A combination of a positional feature and colon and capital features was shown to yield the largest increase in performance. Comparison against the output when using no features at all showed increased in the F-scores from 86.79 to 88.18 (micro-averaged) and 87.08 to 87.85 (macro-averaged) on the regular CRF output. The output after our post-processing step, which was aimed at eliminating potentially misclassified tokens under the class labels *Clinical background and question* and *Conclusion*, showed the overall micro-averaged F-score increasing from 89.03 to 89.30 when using these features, and a slight decrease in the macro-averaged F-score from 88.66 to 88.60.

The classes *Musculoskeletal* and *Armpits* scored lowest on recall, but good on precision. We believe the low recall on *Musculoskeletal* can be explained by the fact that texts labelled under this class were often somewhat integrated within the context. Therefore, the texts could not always be labelled separately from its context (e.g. when the sentence 'No indication for musculoskeletal abnormalities in this area.' occurred in a text on the Abdomen/pelvis). In these cases, we chose to

annotate the text with the context class to preserve meaning. The low recall on the *Armpits* class can be explained by the shortage of training examples. Not all annotated reports contained findings on the armpits area. The same holds for the class *Retroperitoneum*, which scores lowest on precision and reasonable on recall.

We hypothesized that the word frequency features would improve the results in these less frequent classes especially. However, the addition of feature columns that held information on word frequencies under the distinct class labels did not improve results any further. We suspect that by adding the word frequency feature columns, the extra information contained in those features does not extend beyond the increase in extra feature space that is a result of adding the new feature columns, and therefore decreases the results.

The results from the machine learning system support the work of Esuli et al. [16], who used a LC-CRFs learner for the analysis of mammography reports in the Italian language. Their baseline system is comparable to the system used in this study, although our system is trained on longer word strings while these authors trained their system on extracting relatively short pieces of information only (average length of 17.33 words). Also the amount of labels in their study was relatively low with nine distinct labels, resulting in only a few of the important pieces of information being automatically extracted.

For our system or comparable systems to get widely adopted for report processing in the radiological setting, results will need to be further improved. Radiologists will only allow a computational system to automate part of their tasks if they have enough trust and confidence in the system making the correct classifications. To acquire this trust, scores on the classifications will need to be further improved. Radiologists working with the system will not accept classification errors on a regular basis, especially on word sequences or sentences that are easily identifiable (i.e. for humans who have a medical background). For the time being, this means that more work is needed.

A limitation of this study is that we only used reports on the malignant lymphoma. Future research should include reports with a variety of diagnostic questions to ensure that the machine learning system works for any type of report.

The next step for a structured reporting machine learning system would be to extract detailed qualitative and quantitative information from the reports. For example, the system could extract all descriptions and measurements of a specific lesion from multiple reports of the same patient. The fact that our system can classify text in the report is already a good first step in this direction.

In the ideal situation, radiologists have the opportunity to completely focus on image interpretation, without experiencing any distractions from their environment. For the reporting system as a whole, this would mean that the input for further computational steps would be not much more than a single block of text

with findings and a conclusion. The text may even lack any form of structure or punctuation, since radiologists should not have to worry about this. The ideal systems then, would be able to effectively and efficiently process this input into a structured information format that can be shown in real time to the radiologist for direct editing when needed. The structured information could furthermore be converted into a standardized, structured report format specifically suited for (individual) referring clinicians or for other use in the medical environment.

Since medical texts are highly susceptible to abbreviations, poor sentence structure, incorrect spelling and more human-introduced complexities, they are hard to process using conventional language processing systems. The machine learning system of conditional random fields that was used in this study to assign class labels and therefore extract texts in free-text radiological reports for use in further computational systems yielded good results on both larger and smaller classes and showed that machine learning systems provide a good means for classification of information in free-text radiological reports.

Our results, together with further technological advances in information extraction and support systems leave us with promising prospects for the radiology report.

Conclusion

A machine learning system based on the LC-CRF algorithm can be used to automatically structure and standardize the information contained in free-text radiology reports. Such a system would be an interesting alternative to conventional structured reporting systems, because it requires no additional actions from the radiologist and does not interfere with image viewing and interpretation. Our initial performance results with this system are promising. However, more research is needed to improve the performance of the system to a level that is acceptable in clinical practice, apply it to different classes of reports, and extract more detailed information from the reports.

References

[1] Radiological Society of North America. RSNA Radiology Reporting Initiative. http://reportingwiki.rsna.org. Accessed April 3, 2014.

[2] C.P. Langlotz, RadLex: A new method for indexing online educational materials, RadioGraphics 26 (2006) 1595–1597.

[3] American College of Radiology, Illustrated breast imaging reporting and data system (BI-RADS) 3rd ed. Reston, VA: American College of Radiology, 1998.

[4] A.J. Johnson, Radiology report quality: a cohort study of point-and-click structured reporting versus conventional dictation, Acad. Radiol. 9 (2002) 1056–1061.

[5] A.J. Johnson, M.Y. Chen, J.S. Swan, K.E. Applegate, B. Littenberg, Cohort study of structured reporting compared with conventional dictation, Radiology 253 (2009) 74–80.

[6] A.J. Johnson, M.Y. Chen, M.E. Zapadka, E.M. Lyders, B. Littenberg, Radiology report clarity: a cohort study of structured reporting compared with conventional dictation, J. Am. Coll. Radiol. 7 (2010) 501–506.

[7] L. Schwartz, D. Panicek, A. Berk, Y. Li, H. Hricak, Improving communication of diagnostic radiology findings through structured reporting, Radiology 260 (2011) 174–181.

[8] L. Robert, M.D. Cohen, G.S. Jennings, A new method of evaluating the quality of radiology reports, Acad. Radiol. 13 (2006) 241–248.

[9] F. Pool, S. Goergen, Quality of the written radiology report: a review of the literature, J. Am. Coll. Radiol. 7 (2010) 634–643.

[10] S. Wang, R.M. Summers, Machine learning and radiology, Med. Image Anal. 6 (2012) 933–951.

[11] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., Williamstown, MA, 2001: pp. 282–289.

[12] T. Kudo, CRF++: Yet another CRF toolkit, https://taku910.github.io/crfpp/. Accessed November 27, 2013.

[13] H. Cunningham, D. Maynard, K. Bontcheva, Text processing with GATE, Murphys, CA: Gateway Press, 2011.

[14] H. Cunningham, V. Tablan, A. Roberts, K. Bontcheva, Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. PLoS Comput. Biol. 9 (2013) e1002854.

[15] C.D. Manning, P. Raghavan, H. Schutze, Introduction to information retrieval, Cambridge University Press, 2008.

[16] A. Esuli, D. Marcheggiani, F. Sebastiani. An enhanced CRFs-based system for information extraction from radiology reports, J. Biomed. Inform. 46 (2013) 425–435.