

Engineering Natural Language Processing Solutions for Structured Information from Clinical Text: Extracting Sentinel Events from Palliative Care Consult Letters

Neil Barrett^a, Jens H. Weber-Jahnke^a, Vincent Thai^b

^a Department of Computer Science, University of Victoria, Victoria, BC, Canada

^b University of Alberta Hospital, Edmonton, AB, Canada

Abstract

Despite a trend to formalize and codify medical information, natural language communications still play a prominent role in health care workflows, in particular when it comes to hand-overs between providers. Natural language processing (NLP) attempts to bridge the gap between informal, natural language information and coded, machine-interpretable data. This paper reports on a study that applies an advanced NLP method for the extraction of sentinel events in palliative care consult letters. Sentinel events are of interest to predict survival and trajectory for patients with acute palliative conditions. Our NLP method combines several novel characteristics, e.g., the consideration of topological knowledge structures sourced from an ontological terminology system (SNOMED CT). The method has been applied to the extraction of different types of sentinel events, including simple facts, temporal conditions, quantities, and degrees. A random selection of 215 anonymized consult letters was used for the study. The results of the NLP extraction were evaluated by comparison with coded sentinel event data captured independently by clinicians. The average accuracy of the automated extraction was 73.6%.

Keywords:

Natural Language Processing, Palliative Care, Letter, Sentinel Event.

Introduction

Natural language information still plays an important role in health care processes and workflows. For example consultations and referrals often convey important patient-specific health information in form of letters exchanged between providers. Natural language consult letters are not readily machine-interpretable because of their lack of structure and codification. Natural language processing (NLP) attempts to bridge the gap between unstructured and structured information. There is growing interest in using NLP methods in health informatics.

Unique challenges arise with the application of NLP to health care information. For example, clinical narrative is often ungrammatical. General-purpose NLP systems often perform poorly when applied to clinical notes and narratives [1]. Most NLP solutions that perform successfully in the medical domain have been designed and fine-tuned for specific purposes. These solutions have been developed and evolved with great expertise and require significant experience. Unfortunately, the process of developing an NLP solution for medical informatics problems is still more of an art than an engineering science. Guidance on systematic engineering methodologies and re-usable components and processes remain scarce.

The purpose of this paper is twofold. Firstly, we document a systematic method (“blueprint”) for engineering NLP solutions for the medical informatics domain. More precisely, the types of NLP solutions that can be built with our method are purposed for automatically extracting codified information from clinical narrative. Our blueprint includes several novel aspects when compared to other NLP architectures, e.g., it makes use of point-of-speech tagging information during tokenization and it uses ontological knowledge during concept extraction.

Secondly, we present the results of a study evaluating the efficacy of an NLP solution built under this “blueprint” for addressing the problem of different heterogeneous types of sentinel events¹ from palliative consult letters. Sentinel events are of interest to predict survival and trajectory for patients with acute palliative conditions. We measure the accuracy of our NLP solution by comparing its automatically extracted output against two sets of data: (1) a set of physician-extracted sentinel event data, and (2) a set of physician-collected sentinel event data. The difference between the two data sets will be described in detail. Our results indicate a correlation between explicit information content and a high level of accuracy of the automatic extraction method.

Materials and Methods

Generic Method for NLP Information Extraction

The general term of NLP encompasses processing of spoken language (audio) as well as the processing of written text. This paper is concerned with the latter. More precisely, we are interested in NLP problems that attempt to extract structured, codified information from clinical narratives. Figure 1 presents a generic architecture for NLP solutions addressing this class of problems.

In the first two steps, the input text is segmented according to its sentence structure and then individual sentences are further broken down into individual text units referred to as *tokens* [2]. Part-of-speech (POS) tagging assigns speech categories (as tags) to tokens, such as assigning the tag *noun* to the token *thorax*. A POS tag supplies information on its tagged word and on surrounding words. For example, it is likely that a noun follows after the word *the* (e.g., “the hands”), whereas it is less likely that a verb follows the (e.g., “the wrote”) [3].

Using the tagged tokens, the parsing step analyzes the grammatical structure of each sentence, i.e., its syntax [3].

The next processing step (automated clinical coding, ACC) attempts to identify coded concepts in the parsed sentences

¹ A sentinel event is “an unexpected occurrence involving death or serious physical or psychological injury, or the risk thereof” [www.jointcommission.org/SentinelEvents]

and to annotate the sentence structure with these codes [4]. This step makes use of standard clinical terminologies such as SNOMED CT.

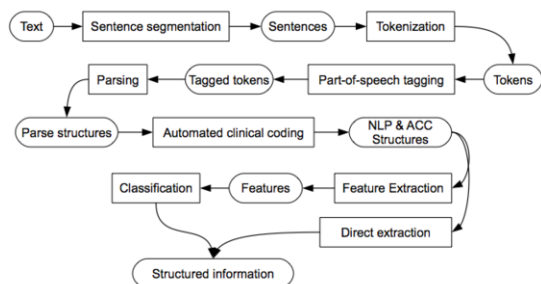


Figure 1- NLP system components

The last processing step extracts the desired information from the parsed and code-annotated text using a hybrid approach involving feature classification and direct (heuristic-based) extraction. The rationale for this hybrid approach is that a classifier cannot extract all types of information that may be required. Direct (heuristic-pattern based) extraction is required, particularly when it comes to extracting quantities, dates, and date range information.

Having given a high-level overview of the components in a generic NLP system used for extracting coded concepts from clinical narrative, we will use the rest of this section to describe the language processing methods we used in more detail. We particularly highlight novel aspects of our method and architecture.

Tokenization and POS tagging method

The accuracies of the tokenization and POS tagging steps have impact on the effectiveness of the downstream NLP processing chain. Despite the importance of tokenization, there is no single widely accepted method for biomedical text, yet neither is biomedical tokenization trivial. Moreover, POS taggers trained for general language communication (e.g. newspaper text) tend to perform poorly when segmenting clinical text [5]. Therefore, a number of dedicated tokenizers and POS taggers have been developed for the biomedical domain, e.g. MedPost [6] and Specialist [7]. Two drawbacks of this approach are (1) the dedicated effort required in developing such domain-specific components, and (2) the relative scarceness of large domain-specific corpora of clinical narratives that can be used to train such dedicated components.

The method we developed for tokenization and POS tagging addresses these drawbacks. Rather than implementing tokenization and POS tagging as two subsequent steps, we utilize feedback between POS analysis and segmentation. The resulting POS-tokenizer performs three major steps:

1. Create a bounded lattice representing a phrase's potential segmentations (cf. Figure 2 for an example).
2. Execute a set of in-place transformations (*transducers*) to normalize elements in the token lattice, e.g. transform them into a canonical form. For example, the string “mg” would be transformed to “milligrams”.
3. Select the segmentation from the token lattice that yields the most probable part-of-speech tagging sequence. We use a Naïve Bayes classifier that considers the current token and its immediate context (previous and next POS tag) for computing the probability of its assigned POS tag.

A formal definition and evaluation of our tokenization and tagging method is out of scope of this paper and can be found

in [5,8]. These earlier experiments indicated that POS tagging is effective in disambiguating tokenization of biomedical text [5]. Moreover, we established that our POS tagger performs significantly more accurately than other leading POS taggers in cross-domain scenario, i.e., if trained on non-biomedical corpora and tested on medical text [5]. We also showed that our POS tagger's accuracy is statistically indistinguishable from the accuracy of other leading methods that have specifically been trained on biomedical corpora [5].

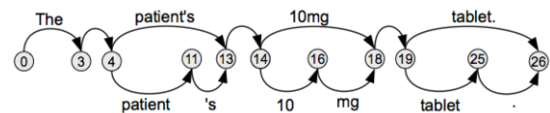


Figure 2- A lattice representing a phrase's segmentation

Parsing method

Context free grammars and dependency grammars are popular methods applied for syntactic parsing. We selected the latter as it is faster and generates a presentation more suitable for subsequent semantic processing of concept relations. MST [9] and Malt [10] were the two best dependency parsers in the “CoNLL-X shared task on Multilingual Dependency Parsing” report [11], which measured dependency parsing performance in 13 languages. As the above competition demonstrated the generalizability of both parsing techniques, we set out to incorporate one of them into our architecture, rather than implementing our own parser from scratch. However, nothing was known about how the performance of MST and Malt would compare in presence of an imperfect input stream of POS tagged tokens. We therefore set up an experiment to investigate this, and found Malt to be more robust to “up-stream” errors of the POS tagged input token sequence. Details have been reported in [8].

Automatic Clinical Coding method

The purpose of the Automatic Clinical Coding (ACC) step is to map segments of the parsed text to codified concepts, as defined in a controlled clinical terminology system. We use the SNOMED CT (SCT) terminology system, as it is widely used internationally and provides a rich set of semantic relationships between the defined terms. Each SCT concept is associated with several natural language descriptions, one of them being marked as the “preferred” description.

Step 1: Tokenization and normalization of the natural language description(s) associated with each concept. The normalization step removes variations among tokens with the same or similar semantics. This includes stemming (e.g., “fractures” is normalized to “fracture”), written numbers (e.g., “two” and “II” become “2”), and abbreviations (e.g., “HIV”). As a result of this step, each SCT concept is associated with a set of normalized tokens², referred to as *semantic atom set*.

Step 2: Pinpoint semantic atoms to concepts where they are first introduced. This step traverses the SCT concept poly-hierarchy and discards tokens from semantic atom sets of concepts with parents that have the same tokens in their semantic atom set.

Step 3: Perform token-level coding. Map each token in the input stream of clinical narrative to the set of SCT concepts where that token appears in the associated semantic atoms set.

Step 4: Combine multiple tokens into valid SCT pre-coordinated and post-coordinated expressions, using the syntactic structure and the POS tags of the input text. This step is

² or several sets if alternative descriptions exist.

done by implementing SCTs rules on constructing valid expressions.

Step 5: Select the most general SCT concept. Multiple concepts may have been mapped to a given linguistic structure. This step selects the most general one according to the SCT hierarchy.

A more formal algorithmic definition of the ACC processing step is given in [8].

Information Extraction

The purpose of the last step Figure 1 is to extract the actual information that is sought as the output of the NLP process. In the case of the study presented in this paper, we sought to extract 17 sentinel events, which we are going to define in more detail in the following section.

The method we used for information extraction (IE) is a hybrid between feature-based classification and direct, template-based extraction. We use support vector machines (SVM) and decision trees as classifiers [12].

Evaluation method

In previous publications, we have evaluated each component of our blueprint architecture in comparison to other leading NLP solution components in the biomedical domain. (See [5,8,13] for details on these component-wise comparison experiments.) This paper will now describe evaluation of the entire NLP system end-to-end in context of a real-world concrete problem, namely that of extracting sentinel events from palliative care consult letters. The set of sentinel events used was taken from an earlier research project of one of the authors (VT). They are listed below:

- Dyspnea [yes/no]
- Dyspnea at rest [yes/no]
- Delirium [yes/no]
- Brain metastases (leptomeningeal) [yes/no]
- Sepsis [yes/no]
- Infection [yes/no]
- Infection site [sites include urinary tract/intra-abdominal/skin]
- Chest infection, aspiration related [yes/no]
- IV antibiotic use [yes/no]
- IV antibiotic use response [no/partial/complete]
- Oral antibiotic use [yes/no]
- Oral antibiotic use response [no/partial/complete]
- Serum creatinine [integer; pattern-based IE]
- Serum creatinine date [date; pattern-based IE]
- Dysphagia [yes/no]
- Previous VTE [yes/no]
- VTE [yes/no]
- ICU Stay [yes/no]
- ICU length of stay in days [integer; pattern-based IE]

Study data

A total of 215 palliative care consult letters were used in this study (200 randomly selected ones, plus 15 “reserved” letters chosen by one of the authors, VT). These were anonymized by hospital staff members prior to be released to the researchers. No other pre-processing of the text contained in the letters was carried out. Two data sets of sentinel events (*physician-extracted* and *physician-collected*) were used to measure the accuracy of the automated NLP extraction process.

The first data set (*physician-extracted*) was specifically created for our study by an expert palliative care physician who manually analyzed the 15 selected letters and extracted the sentinel events of interest. This data set was considered perfectly accurate, i.e., the “gold standard” for the automatic ex-

traction. Due to resource constraints, however, the sample size of the physician-extracted data set is small.

The second data set (*physician-collected*) re-used pre-existing data on sentinel events that were collected earlier for a different study. It was available on the entire sample population of patients (215). These data were collected independently from the consult letters, at a different point in time, and often by a different provider. Consequently, the structured information in the physician-collected data set is not guaranteed to be consistent with the consult letters on the same patient. We will refer to this imperfect nature of the physician-collected data set as the “information gap” analyzed in more detail below.

Feature-based Information Extraction

Most sentinel event IE is treated as a classification task. Each palliative care consult letter is modeled by features and is associated with sentinel event information (e.g., has-sepsis/no-sepsis). A classifier learns the association between features (palliative care consult letters) and labels (sentinel event information). A trained classifier infers omitted labels from a consult letter’s features.

Features are extracted from consult letters as follows. NLP converts text to NLP structures (e.g., dependency graphs). ACC augments these NLP structures with SCT codes. Rather than transforming all SCT encoded linguistic structures into features, linguistic structures are filtered. The post-filter linguistic structures are noun, adjective, preposition and verb phrases. From these linguistic structures, the highest-ranking associated SCT code (if any) becomes a consult letter feature. In negation cases, the feature value is false and omitting the feature implies its absence in the consult letter.

SVMs were trained to extract sentinel event information. A grid search of SVM parameter space established the best parameters for each SVM. In some cases, SVMs overfit the training data, which may result in all tested instances to be classified as the same category, or fewer categories than expected. For example, brain metastases occurs 19 times in 200 consult letters even though an SVM classifies all consult letters as the dominant non-metastases category. When an SVM overfits training data, a decision tree classifier (C4.5, J48) [12] is used instead. The C4.5/J48 decision tree performs pruning in an attempt to not overfit data. SVMs may overfit data due to noise. This is likely the case for sentinel event IE, given that consult letters are modeled with approximately 16,000 features. Decision trees are sensitive to individual features and process features sequentially avoiding some pitfalls experienced by SVMs. For example, in our study SVMs failed to identify true positive examples if few of these examples existed in the data set.

Pattern-based Information Extraction

Sentinel event information such as creatinine levels (integer values) cannot be extracted by trained classifiers. We consequently used pattern-based IE to extract date and integer values. The pattern-based approach is similar regardless of the information being extracted. The approach first locates a *focus* token or SCT code such as creatinine or 15373003. Dates to the left and right of the focus are identified and marked, from tokens closest to the focus to those farthest away. Dates are identified by month. Year and day information is located thereafter. Locating dates excludes numeric date values such as the day (e.g., 21) or the year (e.g., 2009) from being falsely identified as, for example, creatinine levels. Once dates are identified and marked, integer values are identified. The closest date and integer value are bound to the key. In the example sentence “On March 16, 2008, and on March 17, 2008, electrolytes were normal, creatinine was 202.”, creatinine would be bound to the level 202 and the date March 17, 2008.

Extraction and measurement

We performed two evaluation experiments, which are summarized in Figure 3. The first one compared our NLP software's ability to automatically extract sentinel event information to physician-collected sentinel event information (cf. left-hand side of Figure 3). This comparison used a 10-fold cross-validation method. In 10-fold cross-validation, data is randomly split into 10 evenly sized groups. Each group acts as test data while the remaining data (9 groups) act as software training data. For pattern-based IE, the approach was evaluated directly on the 200 consult letters (no training/testing splits). This evaluation used 200 palliative care consult letters paired with their collected sentinel event information for training and testing.

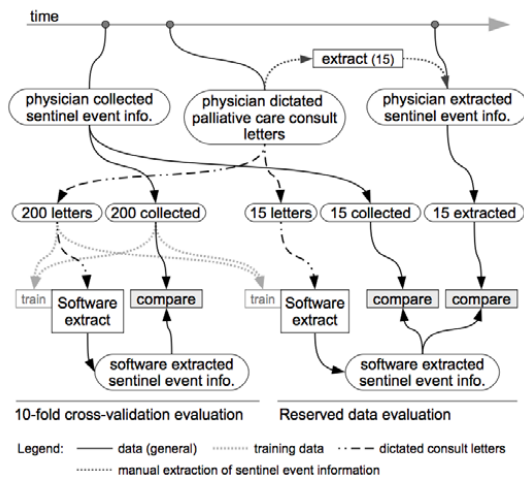


Figure 3- Data creation timeline and use during evaluation

The second evaluation experiment compared our NLP software's ability to automatically extract sentinel event information to physician-extracted and physician collected information on 15 reserved consult letters (cf. right-hand side of Figure 3). The training data consisted for this experiment consisted again of the 200 letters used in the previous experiment. For pattern-based IE, the approach was evaluated directly on the 15 consult letters (no training).

Sentinel Event Info.	Accuracy (%)	C.I. +/-	Avg. f-msr.
Dyspnea	79.5	5.6	70.0
Dyspnea at rest	78.5	5.7	66.3
Delirium	81.0	5.4	79.5
Brain metastases	86.0	4.8	58.2
Sepsis	78.5	5.7	54.2
Infection	50.5	6.9	49.9
Infection site	53.0	6.9	26.5
Chest infection, aspiration related	95.5	2.9	75.1
IV antibiotic use	70.5	6.3	60.3
IV antibiotic use response	57.5	6.9	29.1
Oral antibiotic use	69.5	6.4	45.0
Oral antibiotic use response	68.5	6.4	28.9
Serum creatinine	55.5	6.9	-
Serum creatinine date	37.5	6.7	-
Dysphagia	72.5	6.2	58.3
Previous VTE	93.0	3.5	48.8
VTE	94.0	3.3	49.3
ICU Stay	87.0	4.7	56.1
ICU length of stay in days	90.5	4.1	-
Average	73.6	5.5	53.5

Figure 4- Sentinel event extraction 10-fold cross-validation

Results

The results of our two experiments (accuracy, confidence intervals, and average f-measures) are presented in Figures 4

and 5, respectively. Confidence intervals for 95% confidence were calculated using the normal approximation method of the binomial confidence interval.

In the 10-fold cross-validation experiment on the larger, physician-collected data set, our software's average accuracy was 73.6% with a range of 37.5% to 95.5%. The confidence interval average was +/- 5.5%.

In our second experiment, our software's average accuracy was 78.9% with a range of 53.3% to 100% when evaluated against physician-extracted sentinel event information. When evaluated against physician-collected sentinel event information, the accuracy was 71.9% with a range of 46.7% to 100%. Confidence intervals fell between +/- 0% and +/- 25%.

Sentinel Event Info.	On Extracted	On Collected
Dyspnea	86.7	86.7
Dyspnea at rest	80.0	73.3
Delirium	73.3	66.7
Brain metastases	100.0	100.0
Sepsis	73.3	73.3
Infection	40.0	53.3
Infection site	66.7	53.3
Chest infection, aspiration related	80.0	86.7
IV antibiotic use	66.7	60.0
IV antibiotic use response	53.3	46.7
Oral antibiotic use	73.3	53.3
Oral antibiotic use response	73.3	46.7
Serum creatinine	100.0	66.7
Serum creatinine date	100.0	60.0
Dysphagia	66.7	66.7
Previous VTE	93.3	100.0
VTE	100.0	100.0
ICU Stay	86.7	86.7
ICU length of stay in days	86.7	86.7
Average	78.9	71.9

Figure 5- Sentinel event extraction on reserved data

Discussion

The accuracies measured with our NLP system are similar to state-of-the-art results reported by Stanfill et al. [4] (accuracy range 60% - 100%; precision range 10% - 100%; recall range 25% - 100%).

Information gap

Inconsistencies exist between physician-collected and physician-extracted sentinel event information. Figure 6 quantifies these inconsistencies. For example, collected and extracted delirium information matched 86.7%, respectively differed by (a gap of) 13.3%.

Sentinel Event Info.	Match (%)	Gap (%)	Small Gap
Dyspnea	93.3	6.7	✓
Dyspnea at rest	86.7	13.3	✓
Delirium	86.7	13.3	✓
Brain metastases	100	0	✓
Sepsis	73.3	26.7	
Infection	73.3	26.7	
Infection site	66.7	33.3	
Chest infection, aspiration related	93.3	6.7	✓
IV antibiotic use	93.3	6.7	✓
IV antibiotic use response	60.0	40.0	
Oral antibiotic use	66.7	33.3	
Oral antibiotic use response	53.3	46.7	
Serum creatinine	66.7	33.3	
Serum creatinine date	60.0	40.0	
Dysphagia	100	0	✓
Previous VTE	93.3	6.7	✓
VTE	100	0	✓
ICU Stay	100	0	✓
ICU length of stay in days	93.3	6.7	✓

Figure 6- Match between physician-collected and physician-extracted sentinel event information, over 15 consult letters

Several factors may contribute to the observed gap, including

- *Errors* in data collection or data extraction
- *Quality variations* in consult letters.
- *Purpose variation* of sentinel event information collected for research purpose, while consult letters were written as part of the care workflow. Some detailed information collected for research may rarely be included in a palliative care consult letter, e.g., specific dates.
- *Temporal proximity and relevance*, i.e. consult letters were written *after* the physicians collected the sentinel event data. Some sentinel event information may have been excluded from consult letters because the information may be of secondary importance or could impede readability.

The information gap potentially explains some performance discrepancies when comparing automatically extracted sentinel event information to physician collected information on reserve data. A linear regression produces a $p \leq 3.8^{-5}$. Thus, there is a correlation between the information gap and software performance for physician-collected information (reserve data). This implies that the information gap limits performance on reserve results. Given the correlation's p-value, it is likely the dominant factor affecting this performance.

If the reserve data is a representative sample of the 10-fold data, then the information gap is likely present over all data. If the information gap is present over all data, then it may also explain the 10-fold results. A sample size of 15 is not sufficient (confidence intervals of up to 25%) to simply extend the information gap to the 10-fold data. However, a linear regression between software performance on physician collected reserve results and 10-fold results produces $p \leq 4.9^{-5}$. Thus, the physician collected reserve data is similar to the 10-fold data when using our software as a similarity metric. It is likely that the information gap seen in the reserve data is present in the same form as in the 10-fold data.

The relatively small data set (215 letters) and its imperfect characteristic (information gap) is a limitation of our research result. However, our results indicate the viability of our method and are strong enough to justify resources for making available larger study data sets for further experimentation.

Conclusion

NLP has many potential and beneficial applications in health care, including but not limited to quality control, decision-support, and research. Unfortunately, existing NLP solutions for biomedical applications tend to be idiosyncratic, specialized and expensive to develop. There is a general lack of methodological guidance and reusable components for putting together new NLP solutions in this domain. Another challenge is the relative scarceness of corpora of clinical text needed for training and calibrating such solutions. We have set out to narrow this gap by researching and developing a generic blueprint for NLP solutions used for information extraction from clinical narratives. This blueprint includes an integrated tokenizer / POS tagger component that can be trained on generally available English corpora and performs well when applied to the clinical domain. Moreover, the blueprint implements a novel automated clinical coding (ACC) method that uses the semantic relationships in SCT for assigning codes to linguistic structures.

We have evaluated our method tackling the problem of extracting sentinel events from palliative care consult letters. While the accuracy or our experimental measurements is limited by an imperfection in available study data (see discussion on the "information gap" above), our results are on par with those of other state-of-the-art systems in similar applications.[8]

We expect that our method and blueprint is generalizable and will be useful for constructing similar NLP solutions for information extraction from clinical narratives in an economic and systematic way. Our long-term vision is to make our method available in the form of a medical language processing (MLP) toolkit, similar to the Natural Language Tool Kit (nltk.org), but with specific support for applications in the medical domain.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Components of the Natural Language Tool Kit (nltk.org) were reused in our system. Thanks to Francis Lau for supporting this work and providing valuable comments.

References

- [1] Friedman C and Johnson SB. Natural language and text processing in biomedicine. In E. H. Shortliffe and J. J. Cimino, eds, Biomedical Informatics - Computer Applications in Health Care and Biomedicine, Health Informatics, chapter 8, p. 312–343. Springer NY, 3rd ed, 2006.
- [2] Webster JJ and Kit C. Tokenization as the initial phase in NLP. In Proc. of the 14th Cong on Computational linguistics, p. 1106–1110, Morristown, NJ, USA, 1992. ACL.
- [3] Jurafsky D and Martin JH. Speech and Language Processing. Prentice Hall, Upper Saddle River, NJ, 2009.
- [4] Stanfill MH, Williams M, Fenton SH, Jenders RA, and Hersh WR. A systematic literature review of automated clinical coding and classification systems. J Am Med Inform Assoc, 17(6):646–51, 2010.
- [5] Barrett N and Weber-Jahnke JH. Building a Biomedical Tokenizer Using the Token Lattice Design Pattern and the Adapted Viterbi Algorithm. BMC Bioinformatics 2011, 12(Suppl 3):S1doi:10.1186/1471-2105-12-S3-S1
- [6] Smith L, Rindflesch T, and Wilbur WJ. Medpost: a part-of-speech tagger for biomedical text. Bioinformatics, 20(14):2320–1, Sep 2004.
- [7] NLM. Specialist text tools, 2011
- [8] Barrett N. Natural language processing techniques for the purpose of sentinel event information extraction. PhD Thesis. University of Victoria, Computer Science, 2012.
- [9] McDonald R, Pereira F, Ribarov K, and Hajič J. Non-projective dependency parsing using spanning tree algorithms. Proc. of Conf. on Human Language Technology and Empirical Methods in NLP, p. 523–530. ACL, 2005.
- [10] Nivre J, Hall J, Nilsson J, Eryigit G, and Marinov S. Labeled pseudo-projective dependency parsing with support vector machines. Proc. 10th Conf. on Computational Natural Language Learning, p. 221–225. ACL, 2006.
- [11] Buchholz S and Marsi E. Conll-x shared task on multilingual dependency parsing. 10th Conf. on Computational Natural Language Learning, p. 149–164. ACL, 2006.
- [12] Alpaydin E. Introduction to Machine Learning. MIT Press, Cambridge, MA, USA, 2 ed., 2010.
- [13] Barrett N, Weber-Jahnke JH and Thai V. Automated Clinical Coding using Semantic Atoms and Topology. 25th IEEE CBMS. June 20–22, Rome Italy, 2012

Address for correspondence

Jens Weber, University of Victoria, Department of Computer Science, PO Box 3055, Victoria V8W3P6, Canada BC. Email: jens@uvic.ca