Assessment of Utility in Web Mining for the Domain of Public Health

Peter von Etter, Silja Huttunen, Arto Vihavainen,
Matti Vuorinen and **Roman Yangarber**
Department of Computer Science
University of Helsinki, Finland
First.Last@cs.helsinki.fi

Abstract

This paper presents ongoing work on application of Information Extraction (IE) technology to domain of Public Health, in a real-world scenario. A central issue in IE is the *quality* of the results. We present two novel points. First, we distinguish the criteria for quality: the objective criteria that measure correctness of the system's analysis in traditional terms (F-measure, recall and precision), and, on the other hand, subjective criteria that measure the *utility* of the results to the end-user.

Second, to obtain measures of utility, we build an environment that allows users to interact with the system by rating the analyzed content. We then build and compare several classifiers that learn from the user's responses to predict the relevance scores for new events. We conduct experiments with learning to predict relevance, and discuss the results and their implications for text mining in the domain of Public Health.

1 Introduction

We describe an on-going project for text mining in the domain of Public Health. The aim of the project is to build a system for providing decision support to Public Health (PH) professionals and officials, in the task of Epidemic Surveillance.

Epidemic surveillance may be sub-divided into *indicator-based* vs. *event-based* surveillance, (Hartley et al., 2010). Whereas the former is based on structured, quantitative data, which is collected, e.g., from national or international clinical laboratories or databases, and is of reliable quality, the latter is much more noisy, and relies on "alert and rumour scanning", particularly from *open-source media*, such as on-line news sites. While the latter kind of information sources are less reliable overall, they nonetheless constitute a crucial channel of information in PH. This is because the media are extremely adept at picking up isolated cases and weak signals—which may be indicative of emergence of important events, such as an incipient epidemic or critical change in a public-health situation—and in many cases they can do so much more swiftly than official channels. National and supra-national (e.g., European-level) Health Authorities require timely information about threats posed to the public by emerging infectious diseases and epidemics. Therefore, these Agencies rely on media-monitoring as a matter of routine, on a continual basis as part of their day-to-day operations.

The system described in this paper, PULS, is designed to support Epidemic Surveillance by monitoring open-source media for reports about events of potential significance to Public Health (Yangarber and Steinberger, 2009). We focus in this paper on news articles mentioning incidents of infectious diseases. The system does not make decisions, but provides decision support, by filtering massive volumes of information and trying to identify those cases that should be brought to the attention of *epidemic intelligence officers* (EIO)—public health specialists engaged in epidemic surveillance.

This is an inter-disciplinary effort. The system builds on methods from text mining and computational linguistics to identify the items of potential interest (Grishman et al., 2003). The EIOs, on the other hand, are medical professionals, and are generally not trained in computational methods. Therefore the tools that they use must be intuitive and must

Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents, pages 29–37,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

not overwhelm the user with volume or complexity.

A convenient baseline for comparison is keyword-based search, as provided by search engines and news aggregators. Systems that rely on keyword-matching to find articles related to infectious threats and epidemics quickly overwhelm the user with a vast amount of news items, much of which is noise.

We have tuned PULS, the "Pattern-based Understanding and Learning System," to support Epidemic Surveillance in several phases. PULS is a collaborative effort with MedISys, a system for gathering epidemic intelligence built by the European Commission (EC) at the Joint Research Centre (JRC) in Ispra, Italy. First, MedISys finds news articles from thousands of on-line sources around the world, identifies articles potentially relevant to Epidemic Surveillance, using a broad keyword-based Web search, and sends them via an RSS feed to PULS on a continual basis. Second, PULS employs "fact-finding" technology, *Information Extraction* (IE), to determine exactly what happened in each article: who was affected by what disease/condition, where and when—creating a structured record that is stored in the database. Articles that do not trigger creation of a database record are discarded. A third component then determines the *relevance* of the selected articles—and cases that they describe—to the domain of Public Health, specifically to Epidemic Surveillance.

Traditionally in IE research, performance has been measured in terms of formal *correctness*—how accurately the system is able to analyze the article (Hirschman, 1998). In this paper we argue the need for other measures of performance for text mining, using as a case study our application of Web mining to the domain of Public Health. In the next section, we lay down criteria for judging *quality*, and present the approach taken in our system. Section 3 outlines the organisation of the system, and Section 4 presents in detail our experiments with automatic assignment of relevance scores. In the final section we discuss the results and outline next steps.

2 Criteria for quality

In this section we take a critical view at traditional measures of *quality*, in text analysis in general, and IE in particular. What defines quality most appropriately for our application, and how should we measure quality? We propose the following taxonomy of quality in our context:

- Objective: system's perspective
  - Correctness
  - Confidence
- Subjective: user's perspective
  - Utility or relevance
  - Reliability

At the top level, we distinguish *objective* vs. *subjective* measures. Most IE research has focused on correctness over the last two decades, e.g., in the MUC and ACE initiatives (Hirschman, 1998; ACE, 2004). Correctness is a measure of how accurately the system extracts the semantics from an article of text, in terms of matching the system's answers to a set of answers pre-defined by human annotators. In our context, a set of articles is annotated with a "gold-standard" set of database records, each record containing fields like: the name of the disease/infectious agent, the location/country of the incident, the date of the incident, the number of victims, whether they are human or animal, whether they survived, etc. Then the system's response can be compared to the gold standard and correctness can be computed in terms of recall and precision, F-measure, accuracy, etc.—counting how many of the fields in each record were correctly extracted. This approach to quality is similar to the approach taken in other areas of computational linguistics: how many structures in the text were correctly identified, how many were missed, and how many spurious structures were introduced.

Confidence has been studied as well, to estimate the probability of the correctness of the system's answer, e.g., in (Culotta and McCallum, 2004). Our system computes *confidence* using discourse-level cues, (Huttunen et al., 2002): e.g., confidence decreases as the distance between event trigger and event attributes increases—the sentence that mentions that someone has fallen ill or died is far from the mention of the disease. Confidence also depends on uniqueness of attributes—e.g., if a document mentions only one country, the system has

more confidence that an event referring to this country is correct.

On the subjective side, utility, or relevance, asks how *useful* the result is to the user. There are several points to note. First, it is clearly a highly subjective measure, not easy to capture in exact terms. Second, it is "orthogonal" to correctness in the sense that from the user's perspective utility matters *irrespective* of correctness. For example, an extracted case can be 100% correct, yet have very low utility to the user, (for the task of epidemic surveillance)—a perfectly extracted event that happened too long ago would not matter in the current context. Conversely, every slot in the record may be extracted erroneously, and yet the event may be of great importance and *value* to the user. We focus specifically on relevance vs. correctness.

Given the current performance "ceilings" of 70-80% F-measure in state-of-the-art IE, what does correctness of $x\%$ mean in practice? It likely means that if $x > y$ then a system achieving F-measure x is better to have than one achieving y. But what does it say about *utility*? In the best case, correctness may be correlated with utility, in the worst case it is independent of utility (e.g., if the system happens to achieve high correctness on events from the past, which have low relevance). Since we are targeting a specific user base, the user's perspective must be taken into account when estimating quality, not (only) the system's perspective. This implies the need for *automatic assignment of relevance* scores to analyzed events or documents.

Finally, *reliability* measures whether the reported event is "true". The relevance of extracted fact may be high, but is it credible? Can the information be trusted? We list this criterion for quality for completeness, since it is the ultimate goal of any surveillance process. However, answering this requires a great deal of knowledge external to the system, that can only be obtained by the human user through a detailed down-stream verification process. The system may provide some support for determining reliability, e.g., by tracking the performance of different information sources over time, since the reliability of the facts extracted from an article is related to the reliability of the source. It may be possible to classify Web-based sources according to their credibility; some sources may habitually withhold information (for fear of impact to tourism, trade, etc.); other sites may try to attract readership by exaggerated claims (e.g., tabloids). On the other hand, clearly disreputable sites may carry true information. This measure of quality is beyond the scope of this paper.

3 The System: Background

PULS, the *Pattern-based Understanding and Learning System*, is developed at the University of Helsinki to extract factual information from plain text. PULS has been adapted to analyse texts for Epidemic Surveillance.[1]

The components of PULS have been described in detail previously, (Yangarber and Steinberger, 2009; Steinberger et al., 2008; Yangarber et al., 2007). In several respects, it is similar to other existing systems for automated epidemic surveillance, viz., BioCaster (Doan et al., 2008), MedISys and PULS (Yangarber and Steinberger, 2009), HealthMap (Freifeld et al., 2008), and others (Linge et al., 2009).

PULS relies on EC-JRC's MedISys for IR (information retrieval)—MedISys performs a broad Web search, using a set of boolean keyword-based queries, (Steinberger et al., 2008). The result is a continuous stream of potentially relevant documents, updated every few minutes. Second, an IE component, (Grishman et al., 2003; Yangarber and Steinberger, 2009), analyzes each retrieved document, to try to find events of potential relevance to Public Health. The system stores the structured information about every detected event into a database. The IE component uses a large set of linguistic patterns, which in turn depend on a large-scale public health ontology, similar to MeSH,[2] that contains concepts for diseases and infectious agents, infectious vectors and animals, medical drugs, and geographic locations.

From each article, PULS's pattern matching engine tries to extract a set of incidents, or "facts"—detailed information related to instances of disease outbreak. An incident is described by a set of fields, or attributes: location and country of the incident, disease name, the date of the incident, information about the victims—their type (people, animals, etc.),

[1] puls.cs.helsinki.fi/medical

[2] www.nlm.nih.gov/mesh

number, whether they survived or died, etc.

The result of IE is a populated database of extracted items, that can be browsed and searched by any attribute, according to the user's interests. It is crucial to note that the notion of a user's *focus* or interest is **not** the same as the notion of relevance, introduced above. We take the view that the notion of relevance is shared among the entire PH community: an event is either relevant to PH or it is not. Note also, that this view is upheld by several classic, *human-moderated* PH surveillance systems, such as ProMED-Mail[3] or Canadian GPHIN. User's interest is individual, e.g., a user may have specific geographic, or medical focus (e.g., only viral or tropical illnesses), and given the structured database, s/he can filter the content according to specific criteria. But that is independent of the *shared* notion of relevance to PH. User focus can be exploited for targeted recommendation, using techniques such as collaborative filtering; at present, this is beyond the scope of our work.

The crawler and IE components have been in operation and under refinement for some time. We next build a classifier to assign relevance scores to each extracted event and matched document.

4 Experimental Setup

We now present the work on automatic classification of relevance scores. In collaboration with the end-users, we defined guidelines for judging relevance on a 6-point scale, summarized in Table 1.

Criteria	*Score*
New information, highly relevant	5
Important updates, on-going developments	4
Review of current events, potential risk of disease	3
Historical/non-current events Background information	2
Non-specific, non-factive events, secondary topics, scientific studies hypothetical risk	1
Unrelated to PH	0

Table 1: Guidelines for relevance scores in medical news

[3] www.promedmail.org

Note, the separation between the "high-relevance" scores, 4 and 5, vs. the rest; this split is addressed in detail in Section 4.3.

4.1 Discourse features

It is clear that these guidelines are highly subjective, and cannot be encoded by rules directly. In order to model the relevance judgements, we extracted features—the *discourse features*—from the document that are indicative of, or mappable to, the relevance scores. Discourse features try to capture higher-order information, including complex and longer-range inter-dependencies and clues, involving the physical layout of the document, and deeper semantic and conceptual information found in the document. Some examples of discourse features are:

- *Relative-position*, which is represented by a number from zero to 1 indicating the proportion of the document one needs to read to reach the event text;
- *Disease-in-header* is a binary value that indicates whether the disease is mentioned in the headline or the first two sentences;
- *Disease-to-trigger-distance* indicates how far the disease is from the trigger sentence (same as for confidence computation);
- *Recency* is the number of days between the reported occurrence of the event and the publication date;

We compiled over two dozen discourse-level features. It is clear that the discourse features do not determine the relevance scores, but provide weak indicators of relevance, so that probabilistic classification is appropriate. For example, a higher relative position of an event probably indicates lower relevance, but there are often *news summary* articles that gather many unrelated news together, and may contain very important items anywhere in the article.[4] A feature such as *Victim-named*, stating whether the victim's name is mentioned, often indicates lower-relevance events (obituaries, stories

[4] Due to space limitations, we do not provide a detailed list of the discourse features.

about public personalities, etc.). However, sometimes news articles about disease outbreaks deliberately personify the victims, to give the reader a sense of their background, lifestyle, to speculate about the victims' common circumstances.

We describe two classifiers we have built for relevance. A Naive Bayes classifier (NB) was used as the baseline. We then tried to obtain improved performance with Support Vector Machines (SVM).

4.2 Data

The dataset is the database of facts extracted by the system. The system pre-assigns relevance to each event, and users have the option to accept or correct the system's relevance score, through the User Interface, which also allows the users to correct erroneous fills, e.g., if a country, disease name, etc., was extracted incorrectly by the system.

Along with the users, members of the development team also evaluated a sample of the extracted events, and corrected relevance and erroneous fills. The developers are computer scientists and linguists, whereas the users are medics, and because they interpreted the guidelines differently this had an impact on the results, described in Tables 2 and 5.

"Cleaned data": PULS's user interface also permits users to **correct** incorrect fills in the events (in the two rightmost columns in the tables). This allowed us to obtain two parallel sets of examples with relevance labels: the raw examples, as they were automatically extracted by the system, and the "cleaned" examples, after users/developer corrections. The raw set is more noisy, since it contains errors introduced by the system. We used the cleaned examples to train our classifiers, and tested them on both the cleaned set and the raw set. Testing against the cleaned set gives an "idealized" performance, (as if the IE system made no errors in analysis). True performance is expected be closer to testing on the raw set.

In total, there were just under 1000 examples labeled by the users and the developers (some examples were labeled by both, since the system allows multiple users to attach different relevance judgements to the same example. Most of the time users agreed on the relevance judgements, but non-developers were less likely to clean examples.)

4.3 Naive Bayes classifier

Initially, we planned to perform regression to the complete [0–5] relevance scale. However, this proved problematic, since the amount of labeled data was not sufficient to cover the continuum between highly relevant and not-so-relevant items. We therefore decided instead to build a *binary* classifier. This decision is also justified in the context of our system's user interface, which provides the users with two views:

- the *Front Page View* contains only high-relevance items (rated 4 or 5), in case the user wants to see only the most urgent items first;
- the *Complete View* shows the user all extracted items, irrespective of relevance. (The user can always filter the database by relevance value.)

Thus, the relevance score is also used to guide a *binary* decision: whether to present a given event/article to the user on the Front-Page View. The NB classifier using the entire set of discourse features did not perform well, because the discourse features we have implemented are inherently not independent, which affects the performance of NB.

To try to reduce the mutual dependence among the features, we added a simple, *greedy* feature-selection phase during training. Feature selection starts by training a classifier on the full set of features, using leave-one-out (LOO) cross-validation to estimate the classifier's performance. In the next phase, the algorithm in turn excludes the features one by one, and runs the LOO cross-validation again, once with each feature excluded. The feature whose exclusion gives rise to the biggest increase in performance is dropped out, and the selection step is repeated with the reduced set of features. We continue to drop features until performance does not increase for several iterations; in our experiments, we used three steps beyond the top performance. We then back up to the step that yielded peak performance. The resulting subset of features is used to train the final NB classifier.

The NB classifier is implemented in R Language.

Because relevance prediction is difficult for all events, we also tried to predict the relevance of an article, making the simplifying assumption that the article is only as relevant as the *first* event found in

the article.[5] The results are presented in Table 2. The rows labeled *Dev only* refer to the data sets labeled by developers, and *Users only* to sets labeled by (non-developer) users.

	Testing on		Number examples	
	Clean	Raw	Clean	Raw
Event-level				
Dev only	76.96	**76.66**	560	510
All	72.19	73.34	863	799
Users only	70.38	66.53	303	289
Document-level				
Dev only	80.41	**79.00**	291	281
All	73.94	72.45	545	530
Users only	65.82	67.09	238	232

Table 2: Naive Bayes prediction accuracy

The event-level classification is shown in the top portion of the table. Throughout, as expected, testing on the cleaned data usually gives slightly better (more idealized) performance estimates than testing on the raw. Also, as expected, testing on the first-only events (document-level) gives slightly better performance, since it's a simpler problem—although there is less data to train/test on.

It is important to observe that using data labeled by developers gives significantly higher performance. This is because coercing the users to follow the guidelines strictly is not possible, and they deviate from the rules that they themselves helped articulate. The rows labeled "all" show performance when all combined available data was used—labeled by both the developers and the users.

This performance is quite good for a baseline.[6] The confusion matrices—for the developer-only event-level raw data set—show the distribution of true/false positives/negatives.

4.4 SVM Classifier

For comparison, we built two additional classifiers using the SVMLight Toolkit.[7] We first used a linear kernel as a baseline, and used a RBF kernel, which is potentially more expressive. The conditions for testing the SVM classifiers were same as the ones for the NB classifiers, and same datasets were used as for the NB.

	True Labels	
Predicted labels	4-5	0-3
High-relevance 4-5	125	77
Low-relevance 0-3	42	266

Table 3: NB confusion matrix

As SVM with the RBF kernel can use non-linear separating hyperplanes in the original feature space by using the kernel trick (Aizerman et al., 1964), we aimed to test whether it would provide an improvement over the linear kernel. (For more detailed discussions of SVM and different kernel functions for text classification, cf., for example, (Joachims, 1998).)

To regularize the input for SVM, all feature values were normalized to lie between 0 and 1 (for continuous-valued features), and set to 0 or 1 for binary features. Table 4 describes the accuracy achieved with the linear kernel. Experiments labeled *All discourse features* use the complete set of discourse features (over 20 features). Rows labeled *Selected discourse features* show results from training with exactly same features as resulted from the feature selection phase of NB.

	Event-level		*Document-level*	
	Clean	Raw	Clean	Raw
All discourse features				
Dev only	75.33	**77.17**	76.87	76.56
All	71.60	72.26	70.51	69.96
Selected discourse features only				
Dev only	76.07	**77.95**	77.94	77.62
All	71.40	72.14	69.75	69.37

Table 4: SVM prediction accuracy using linear kernel

The difference when training with selected discourse features and all discourse features is not large, since SVM is able to distinguish between relevant and non-relevant features fairly well. The results from SVM using linear kernel appear compa-

[5] A manual check confirmed that there were *no* instances where the first event in an article had lower relevance than a subsequent event.

[6] Consider for comparison, that the *correctness* on a manually constructed, non-hidden set of articles used for system development, is under 75% F-measure.

[7] http://svmlight.joachims.org/

rable with the results from the NB.

In addition to using the discourse features, we also tried using *lexical features*. The lexical features for a given example—extracted event—is simply the bag of words from the sentence containing the event, plus the two surrounding sentences. To reduce data sparsity, the sentences are pre-processed by a lemmatizer, and passed through a named entity (NE) recognizer (Grishman et al., 2003), which replaces persons, organizations, locations and disease names with a special token indicating the NE's class. "Stop-word" parts of speech were dropped—prepositions, conjunctions, and articles.

	Event-level		*Document-level*	
	Clean	Raw	Clean	Raw
All discourse features				
Dev only	74.69	**75.37**	77.93	78.38
All	69.58	70.26	71.56	71.25
Selected discourse features only				
Dev only	77.51	**79.01**	79.19	79.04
All	72.02	72.84	72.59	72.30
Lexical features only				
Dev only	75.93	**76.37**	79.11	80.07
All	73.28	73.47	74.53	74.71
Lexical and selected discourse features				
Dev only	78.87	**79.24**	82.66	81.83
All	76.48	76.58	76.52	76.19

Table 5: SVM prediction accuracy using RBF kernel

The performance of SVM with the RBF kernel is strongly dependent on the values of SVM parameters C—the trade-off between training error and margin— and γ—the kernel width (Joachims, 1998). We tuned these parameters manually by checking a grid of values against a development dataset, and finding areas where the SVM performed well. These areas were then further investigated. After trying 40 combinations, we set C as 10000 and γ to 0.001 for subsequent evaluations. The results for SVM using RBF kernel are given in Table 5.

High accuracy of lexical features alone was somewhat surprising as lexical features consist only of the bag of words in the event-bearing sentence, plus the preceding and the following sentences. News articles often have various pieces of information related to the event scattered around the document. For example, the disease can appear only in the headline, the location/country in the middle of the document, and the event-bearing sentence in a third location, (Huttunen et al., 2002). Our lexical features, as presented here, are not capable of capturing such long-distance relationships.

The observed difference in performance on relevance prediction between the data sets labeled by developers vs. non-developer users, likely arises from the fact that developers follow the formal guidelines more strictly (being computer scientists). Rows labeled *all* show performance against data sets labeled by real users, who work in different PH organizations in several different countries, each group of users intuitively following their own, subjective guidelines, *despite* the common guidelines agreed-upon for this project. There may also be deviation within organizations. For example, certain doctors may find specific diseases or locations more interesting, giving events containing them a high relevance, thus injecting personal preference into document relevance.

5 Discussion and Conclusions

The SVM performs somewhat better than the Naive Bayes classifier, though there is still much to be explored and improved. One odd effect is that sometimes testing on the raw data gives slightly better results than testing on the clean data, though this is probably not significant, since the SVM classifier is still not finely tuned (and the data contain some noise). Using all discourse features performs slightly worse than using a reduced set of features—the same set of features that we obtained through greedy feature selection for NB.

Although the lexical features alone seem to do somewhat worse than the discourse features alone on event-level classification, we still see that the lexical features contain a great deal of information (which the NB cannot use). As expected, adding the discourse features improves performance over lexical features alone, since discourse features capture information about long-range dependencies that local lexical features do not.

In forming splits for cross-validation or LOO, we made sure not to split examples from the same doc-

ument across the training and test sets. That is, for a given document, *all* events in it are either used for training or for testing, to avoid biasing the testing.

To summarize, the points addressed in this paper:

- We have presented a language-technology-based approach to a problem in Public Health, specifically the problem of event-based epidemic surveillance through monitoring on-line media.
- The user's perspective needs to be taken into account when estimating quality, not just the system's perspective. *Utility* to the user is at least as important as (if not more important than) correctness.
- We have presented an operational system that suggests articles potentially relevant to the user, and assigns relevance scores to each extracted event.
- For now, we assume the users share same notion of relevance of an event to Public Health.
- We have presented experiments and an initial evaluation of assignment of relevance scores.
- Experiments indicate that relevance appears to be a *tractable* measure of quality, at least in principle. Marking document-level relevance—only for the first event in the document—appears to be easier. However, making real users follow strict guidelines is difficult in practice.

On-going work includes refining the classification approaches, especially, using Bayesian networks, regression, using transductive SVMs to leverage unlabeled data, and exploring collaborative filtering to address users' individual interests.

Acknowledgments

This research was supported in part by: the Technology Development Agency of Finland (TEKES), through the ContentFactory Project, and by the Academy of Finland's National Centre of Excellence "Algorithmic Data Analysis (ALGODAN)."

References

ACE. 2004. Automatic content extraction.

M. A. Aizerman, E. A. Braverman, and L. Rozonoer. 1964. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control*, volume 25, pages 821–837.

Aron Culotta and Andrew McCallum. 2004. Confidence estimation for information extraction. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics*.

Son Doan, Quoc Hung-Ngo, Ai Kawazoe, and Nigel Collier. 2008. Global Health Monitor—a web-based system for detecting and mapping infectious diseases. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.

C.C. Freifeld, K.D. Mandl, B.Y. Reis, and J.S. Brownstein. 2008. HealthMap: Global infectious disease monitoring through automated classification and visualization of internet media reports. *Journal of American Medical Informatics Association*, 15:150–157.

Ralph Grishman, Silja Huttunen, and Roman Yangarber. 2003. Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, **35**(4):236–246.

David Hartley, Noele Nelson, Ronald Walters, Ray Arthur, Roman Yangarber, Larry Madoff, Jens Linge, Abla Mawudeku, Nigel Collier, John Brownstein, Germain Thinus, and Nigel Lightfoot. 2010. The landscape of international event-based biosurveillance. *Emerging Health Threats Journal*, 3(e3).

Lynette Hirschman. 1998. Language understanding evaluations: Lessons learned from muc and atis. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, pages 117–122, Granada, Spain, May.

Silja Huttunen, Roman Yangarber, and Ralph Grishman. 2002. Complexity of event structure in information extraction. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, August.

Thorsten Joachims. 1998. Text categorization with suport vector machines: Learning with many relevant features. In *ECML: European Conference on Machine Learning*, pages 137–142.

J.P. Linge, R. Steinberger, T.P. Weber, R. Yangarber, E. van der Goot, D.H. Al Khudhairy, and N.I. Stilianakis. 2009. Internet surveillance systems for early alerting of health threats. *Eurosurveillance Journal*, 14(13).

Ralf Steinberger, Flavio Fuart, Erik van der Goot, Clive Best, Peter von Etter, and Roman Yangarber. 2008.

Text mining from the web for medical intelligence. In Domenico Perrotta, Jakub Piskorski, Franoise Soulié-Fogelman, and Ralf Steinberger, editors, *Mining Massive Data Sets for Security*. OIS Press, Amsterdam, the Netherlands.

Roman Yangarber and Ralf Steinberger. 2009. Automatic epidemiological surveillance from on-line news in MedISys and PULS. In *Proceedings of IMED-2009: International Meeting on Emerging Diseases and Surveillance*, Vienna, Austria.

Roman Yangarber, Clive Best, Peter von Etter, Flavio Fuart, David Horby, and Ralf Steinberger. 2007. Combining information about epidemic threats from multiple sources. In *Proceedings of the MMIES Workshop, International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria, September.