Combining HCI, Natural Language Processing, and Knowledge Discovery - Potential of IBM Content Analytics as an Assistive Technology in the Biomedical Field

Andreas Holzinger[1], Christof Stocker[1], Bernhard Ofner[1], Gottfried Prohaska[2],
Alberto Brabenetz[2], and Rainer Hofmann-Wellenhof[3]

[1] Medical University Graz, A-8036 Graz, Austria
Institute for Medical Informatics, Statistics & Documentation,
Research Unit HCI4MED, Auenbruggerplatz 2/V, A-8036 Graz, Austria
{a.holzinger,c.stocker,b.ofner}@hci4all.at
[2] IBM Austria, Obere Donaustraße 95, A-1020 Vienna, Austria
{gottfried_prohaska,a.brabenetz}@at.ibm.com
[3] LKH-University Hospital Graz
Department for Dermatology, Auenbruggerplatz 22/V, A-8036 Graz, Austria
rainer.hofmann@medunigraz.at

Abstract. Medical professionals are confronted with a flood of big data most of it containing unstructured information. Such unstructured information is the subset of information, where the information itself describes parts of what constitutes as significant within it, or in other words - structure and information are not completely separable. The best example for such unstructured information is text. For many years, text mining has been an essential area of medical informatics. Although text can easily be created by medical professionals, the support of automatic analyses for knowledge discovery is extremely difficult. We follow the definition that knowledge consists of a set of hypotheses, and knowledge discovery is the process of finding or generating new hypotheses by medical professionals with the aim of getting insight into the data. In this paper we present some lessons learned of ICA for dermatological knowledge discovery, for the first time. We follow the HCI-KDD approach, i.e. with the human expert in the loop matching the best of two worlds: human intelligence with computational intelligence.

Keywords: Knowledge discovery, data mining, human-computer interaction, medical informatics, Unstructured Information Management, Content Analytics.

1 Introduction and Motivation for Research

Electronic patient records (EPR) contain increasingly large portions of data which has been entered in non-standardized format, which is often and not quite correctly called *free text* [1, 2]. Consequently, for many years, text mining was

A. Holzinger and G. Pasi (Eds.): HCI-KDD 2013, LNCS 7947, pp. 13–24, 2013.

and is an essential area of medical informatics, where researchers worked on statistical and linguistic procedures in order to dig out (mine) information from plain text, with the primary aim of gaining information from data. Although text can easily be *created* by medical professionals, the support of (semi-) automatic analyses is extremely difficult and has challenged researchers for many years [3–5]. The next big challenge is in Knowledge Discovery from this data. Contrary to the classical text mining, or information retrieval approach, where the goal is to find information, hence the medical professional knows what he wants, in knowledge discovery we want to discover novel insights, get new knowledge which was previously unknown. To reach this goal, approaches from pure computer science alone are insufficient, due to the fact that the "real" intelligence is in the brains of the professionals; the next step consists of making the information both usable and useful. Interaction, communication and sensemaking are still missing within the pure computational approaches [6].

Consequently, a novel approach is to combine HCI & KDD [7] in order to enhance human intelligence by computational intelligence. The main contribution of HCI-KDD is to *enable* end users to *find and recognize* previously unknown and potentially useful, usable, and interesting information. Yet, what is interesting is a matter of research [8]. HCI-KDD may be defined as the process of identifying novel valid, and potentially useful data patterns, with the goal to understand these patterns [9]. This approach is based on the assumption that the domain expert possesses explicit domain knowledge and by enabling him to interactively look at his data sets, he may be able to identify, extract and understand useful information, as to gain new - previously unknown - knowledge [10].

1.1 The Challenges of Text

Text, seen as transcription of natural language, poses a lot of challenges for computational analysis. Natural language *understanding* is regarded as an AI-complete problem [14]. In analogy to NP-completeness from complexity theory this means that the difficulty of the computational problem is equivalent to designing a computer which is as intelligent as a human being [15], and which brings us back to the very roots of the computational sciences [16].

It became evident over the past decades that the understanding of human language requires extensive knowledge, not only about the language itself, but also about the surrounding real world, because language is more than words, and meaning depends on context, and "understanding" requires a vast body of knowledge about this real world context [14] - we call this context-awareness [17]. Consequently, natural language processing (NLP) is a term that does not necessarily target a total understanding of language per se [18].

2 Theory and Background

In this section we define and describe the basic notions we use throughout this paper. Our understanding of terms such as "data" is a little uncommon, but based on what we think are good reasons. One being that natural language understanding is an AI-complete problem [14]. It makes more sense to ground the semantics of those terms closer to human understanding rather than "traditional" computer models.

2.1 Unstructured Information Management

There is still no clear definition given so far and the few definitions are very ambiguous (see a recent work as a typical example: [19]).

First, let us define our understanding of certain basic terms; namely data, information, and knowledge. These terms are interpreted quite differently throughout the scientific literature [20].

We ground our definitions on Boisot & Canals (2003) [21] and their sources. They describe **data** as originating in discernible differences in physical states-of-the-world, registered through stimuli. Theses states are describable in terms of space, time, and energy. Significant regularities in this data - whatever one qualifies as significant - then constitutes **information**. This implies that the information gained from data, depends on the agent extracting it - more precisely: his expectations, or *hypotheses*. This set of hypotheses held by an agent can then be referred to as **knowledge** and is constantly modified by the arrival of information.

Definition 1. *If knowledge consists of a set of hypotheses, then* ***knowledge discovery*** *is the process of finding or generating new hypotheses out of information.*

Since what qualifies as significant depends on the agents individual disposition, information can only appear to be *objective*, if what constitutes as significant regularities is established through convention [21].

It might be interesting to note, that based on these definitions, the commonly used term "*unstructured data*" refers to complete randomness, or noise.

Unstructured information, on the other hand, often refers to natural language, be it in the form of written documents, speech, audio, images or video. This implicit definition makes sense, as it is used to split information into two easily understood classes: databases content and everything else. The reason for this is mostly business motivated, as the term "unstructured" is then used to convey the message of computational inaccessibility through information retrieval methods to the "stored" information, and hence a necessity for action.

Let us state a more precise definition:

Definition 2. ***Unstructured Information*** *is the subset of information, where the information itself describes parts of what constitutes as significant regularity.*

What this essentially means, is that *information* and its *structure* are not completely separable. The best example for unstructured information is in text. The meaning of the text - its nouns, verbs, markers and so fourth - partly depends on the text itself - on the context and discourse. Even for humans it can be difficult. Sometimes sentences have to be re-read to be understood, or are misunderstood completely. While processing text, our knowledge-base is constantly being updated by the text itself, and a combination of our previous knowledge and updated knowledge is used to overcome and interpret uncertainties.

2.2 Model Structure through Annotation

In linguistics, the term annotation most commonly refers to meta data used to describe words, sentences and so fourth. The process of annotation is often described as *tagging*, which is the automatic assignment of descriptors to input tokens [22]. One prominent example is part-of-speech (POS) tagging, which maps a natural language, such as english, to a meta-language made up of word classes, such as nouns, verbs, adverbs and so fourth. We will describe POS tagging in more detail later.

The idea behind tagging is to model human knowledge, be it about language or another topic, in a computationally understandable way, in order to help the computer process unstructured information. This is usually not a trivial task, and its complexity strongly depends on the language, domain and the quality of the text. An universal automatic solution would be desirable, but for now seems out of reach.

However, it is our opinion, that by efficiently including the human in the loop, the research progress of computational techniques can vastly be improved. On a business perspective, a good user interface for the developers significantly speeds up domain specific solutions for unstructured information management.

2.3 On the Origins of IBM Content Analytics

For many years, IBM research groups from various countries are working on the development of systems for text analysis, and text-mining methods to support problem solving in life science. The best known system today is called Biological Text Knowledge Services and integrates research technologies from multiple IBM research labs. BioTeKS is the first major application of the so-called Unstructured Information Management Architecture (UIMA) initiative [23]. These attempts go back to a text mining technology called TAKMI (Text Analysis and Knowledge MIning), which has been developed to acquire useful knowledge from large amounts of textual data - not necessarily focused on medical texts [24].

The subsystem of UIMA is the Common Analysis System (CAS), which handles data exchanges between the various UIMA components, including analysis engines and unstructured information management applications. CAS supports data modeling via a type system and is independent of any programming language. It provides data access through a powerful indexing mechanism, hence provides support for creating annotations on text data [25].

BioTeKS was originally intended to analyze biomedical text from MEDLINE abstracts, where the text is analyzed by automatically identifying terms or names corresponding to key biomedical entities (e.g., proteins, drugs, etc.) and concepts or facts related to them [26]. MEDLINE has been often used for testing text analytics approaches and meanwhile a large number of Web-based tools are available for searching MEDLINE. However, the non-standardized nature of text is still a big issue, and there is much work left for improvement. A big issue is in end-user centred visualisation and visual analytics of the results, required for the support of the sensemaking processes amongst medical professionals [27,28].

3 Related Work

Many solutions for data analytics are available either as commercial or open-source software, ranging from programming languages and environments providing data analysis functionality to statistical software packages to advanced business analytics and business intelligence suites.

Prominent tools focusing on statistical analysis are IBM SPSS, SAS Analytics as well as the open-source R project for statistical computations. Each of the aforementioned tools provides additional packages for text analysis, namely IBM SPSS Modeler, a data mining and text analytics workbench, SAS Text Analytics and the tm package for text mining in R.

Software focusing on text mining and text analysis like the Apache UIMA project or GATE (General architecture for text engineering) are aimed at facilitating the analysis of unstructured content. Several projects based on the UIMA framework provide additional components and wrappers for 3rd-party tools, with the purpose of information extraction in the biomedical and the healthcare domain, including Apache cTAKES (clinical Text Analysis and Knowledge Extraction System) and the BioNLP UIMA Component Repository.

Other solutions for knowledge analysis utilize machine learning algorithms and techniques, with the most prominent frameworks Weka (Waikato Environment for Knowledge Analysis) and RapidMiner.

To our knowledge there are only a few publications concerning the integration of UIMA into clinical routine:

Garvin et al. (2012) [29] built a natural language processing system to extract information on left ventricular ejection fraction, which is a key component of heart failure, from "free text" echocardiogram reports to automate measurement reporting and to validate the accuracy of the system using a comparison reference standard developed through human review. For this purpose they created a set of regular expressions and rules to capture "ejection fraction" using a random sample of 765 echocardiograms. The authors assigned the documents randomly on two sets: a set of 275 used for training and a second set of 490 used for testing and validation. To establish a reference standard, two independent experts annotated all documents in both sets; a third expert resolved any incongruities. The test results for documentlevel classification of EF of $< 40\%$ had a sensitivity (recall) of 98.41%, a specificity of 100%, a positive predictive value (precision) of 100%, and an F measure of 99.2%. The test results at the concept level had a sensitivity of 88.9% (95% CI 87.7% to 90.0%), a positive predictive value of 95% (95% CI 94.2% to 95.9%), and an F measure of 91.9% (95% CI 91.2% to 92.7%) - consequently, the authors came to the conclusion that such an automated information extraction system can be used to accurately extract EF for quality measurement [29].

4 Methods

In our project, we are using IBM Content Analytics (ICA) Studio 3.0, which utilizes a UIMA pipeline as depicted in Fig. 1. This pipeline includes a set of fundamental annotators. Note that the first two can not be changed, while the other can be configured according to ones needs.

Language Identification Annotator. Identifies the language of the document. This fundamental information can be utilized to branch in specialized parsing rule sets.

Linguistic Analysis Annotator. Applies basic linguistic analysis, such as POS, to each document.

Dictionary Lookup Annotator. Matches words from dictionaries with words in the text. Note that stemming, as well as the definition of synonyms is supported.

Named Entity Recognition Annotator. This annotator can only be activated or deactivated and not configured as of now. It extracts person names, locations and company names

Pattern Matcher Annotator. Identifies those pattern in the text that are specified via rules, e.g. in the ICA Studio

Classification Module Annotator. Performs automatic classification. It uses natural language processing as well as semantic analysis algorithms to determine the true intent of words and phrases. It combines contextual statistical analysis with a rule-based, decision-making approach

Custom Annotator. Custom annotators are essentially java programs that obey a given UIMA interface. This program has access to all annotations made by the previous annotators.

The interesting idea behind ICA Studio, the development suit behind ICA, is to efficiently include the human in the loop. It does that by offering the developer a quick and easy way to model his knowledge into an UIMA conform format.

Through a mixture of dictionaries, regular expressions and parsing rules, annotation schemes can quickly be realized and instantly tested. With the possibility to plugin custom annotators (Java, or C++ programs) into the UIMA pipeline, custom methods can easily be included and are able to access the realized annotation schemes. This opens a quick and easy way to fast prototyping and modular development, both while offering quality control.

In the following subsections we will describe and discuss some common computational methods that are in one way or another utilized by ICA.

4.1 Morphology

Most natural languages have some system to generate words and word forms from smaller units in a systematic way [22, 30]. The seemingly infinity of words

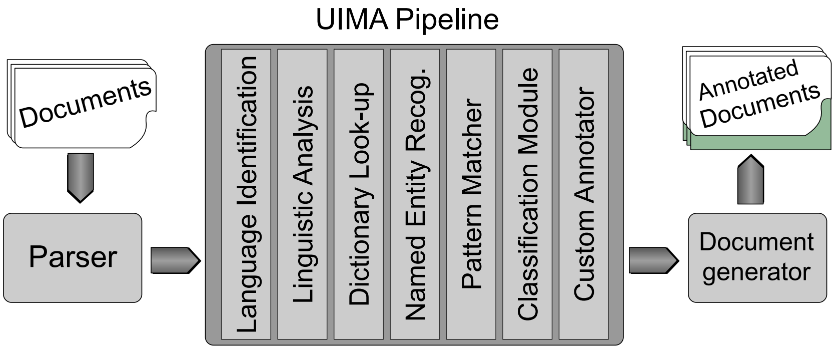

Fig. 1. The document processor architecture in IBM Content Analytics

in a language is produced by a finite collection of smaller units called *morphemes*. Simply put, morphology deals with the structure of words. These morphemes are either semantic concepts like *door*, *house*, or *green*, which are also called roots, or abstract features like *past* or *plural* [22]. Their realization as part of a word are then called *morph*, such as *door* or *doors*.

The information expressed with morphology varies widely between languages. In Indo-European languages for example, distinct features are merged into a single bound form [22]. These languages are typically called *inflectional languages*. Inflections do not change the POS category, but the grammatical function. Inflections and derivations convey information such as tense, aspect, gender or case.

When defining custom dictionaries, ICA Studio supports the manual definition of any custom morph or synonym, as well as providing the automatic generation of inflections. The canonical form is then defined as the *lemma*. For many supported languages, such as German and English, standard dictionaries are already build-in.

Custom dictionaries are most useful to define specific classes. For example a dictionary called “Defects”, that includes various words (and their morphs) indicating a defect, such as “break”, “defect”, or “destroy”, could be used in combination with other annotations to detect faulty products out of customer reports or forum entries.

4.2 Lexicography

The term “*computational lexicography*” can have different meanings. Hanks [22] listed two common interpretations:

1. Restructuring and exploiting human dictionaries for computational purposes
2. Using computational techniques to compile new dictionaries

In this paper we refer to the exploiting of human dictionaries for computational purposes.

The creation of dictionaries in ICA is useful to create word classes. For example a dictionary “MonthNames” could be used to identify mentioning of months within a text documents. The dictionaries also offer the possibility to associate other information with other features. For example “Oktober” could then be associated with “10” to in turn easily normalize date information.

The process of creating a dictionary is simply and supports the automatic generation of inflections, as well as the manual definition of synonyms if so desired.

4.3 Finite-State Technology

Many of the basic steps in NLP, such as tokenization and morphological analysis, can be carried out efficiently by the means of finite-state transducers [22]. These transducers are generally compiled from *regular expressions*, which is a formal language for representing sets and relations [22].

ICA studio utilizes regular expressions to realize character rules. This is useful to specify character patterns of interest such as dates or phone numbers. Those character sequences following these patterns can then be annotated and that way be used accordingly for relation extraction.

4.4 Text Segmentation

Text segmentation is an important step in any NLP process. Electronic text in its raw form is essentially just a sequence of characters. Consequently it has to be broken down into linguistic units. Such units include *words*, *punctation*, *numbers*, *alphanumerics*, etc. [22]. This process if also referred to as *tokenization*. Most NLP techniques also require the text to be segmented into sentences and maybe paragraphs as well [22].

ICA has this functionality included. Beside tokenization, which can be influenced by the means of special dictionaries or character rules, ICA splits the text into sentences and paragraphs. It migh be interesting to note, that neither character rules, nor dictionaries have to influence tokenization, it is a matter of choice.

4.5 Part-of-Speech Tagging

Most tasks NLP require the assignment of classes to linguistic entities (tokens) [30]. Part-of-Speech (POS), for instance, is an essential linguistic concept in NLP, and POS tagger are used to assign syntactic categories (e.g. noun, verb, adjective, adverb, etc.) to each word [30, 31].

Automatic part-of-speech taggers have to handle several difficulties, including the ambiguity of word forms in their part-of-speech [32], as well as classification problems due to the ambiguity of periods ('.'), which can be either interpreted as part of a token (e.g. abbreviation), punctuation (full stop), or both [30].

ICA provides a POS tagger as part of the Linguistic Analysis annotator included in the UIMA Pipeline, and can overcome aforementioned classification difficulties by integration of the user. Default tagging can in later stages be influenced and improved by defining own types, such as real numbers or dates, by means of character rules for the disambiguation of punctations; custom dictionaries allow the user to assign the part-of-speech to special words or word classes - if needed - which can be later exploited in the process of designing parsing rules.

4.6 Information Extraction

The build-in tools of ICA follow the idea of information extraction (IE). Grishman (2003) [22, 30] defines IE as the process of automatically identifying and classifying instances of entities, relations and events in a text, based on some semantic criterion.

Typical task are *name-*, *entity-*, *relation-* and *event extraction* [30]. The first two are integrated into the *named entity annotator*, while the others can be defined within the studio.

The modeling of relations can be done by defining parsing rules, that can operate on different levels, such as phrase or entity, as well as different scopes, such as sentence, paragraph or document. These parsing rules can also automatically be derived out of sample text passages and manually changed and optimized as needed, speeding up the process.

An interesting functionality included in the ICA studio are the so called *normalizers*. They can be used to convert different formats of the same concept into one standardized format. For example: “12.10.1987” and “1987-10-12” describe the same concept - a date. The normalizers can also be used to overcome different points of reference, or units. For example: “100 pounds” could automatically be tagged with the normalized feature “45.359 kg”.

5 Materials

In our experiment, we are interested to investigate the potential to support a medical doctor (dermatologist) in his research by applying NLP techniques to selected *medical sample records*. These records contain a number of structured information, such as *patient name* and *date of birth*, but most of it is unstructured information in the form of written text.

The big challenge when confronted with text written by medical doctors, is the large portion of abbreviations, that in turn are not standardized either.

The idea is to apply IE techniques, in order to associate patients with extracted information such as *diagnosis*, *therapies*, *medicaments*, changes in *tumor size*, and so fourth. These information are then used to create a chronological sequence to enable the medical professional to see trends and correlations - or in other words: apply *knowledge discovery*.

6 Lessons Learned

Realizing an annotation scheme is an iterative task. Even though we as humans know what information is important to us, for example the type and length of a therapy, formulating a rule to identify such information is not a trivial undertaking.

The simple and easy-to-use interface enables the developers to perform fast prototyping. The on-the-fly testing gives you fast feedback of the "quality" of your rules. On the other hand, a lot of basic - and needed - functionality, such as POS tagging and named entity identification, is already build in.

Currently we are investigating different ways to realize our annotation schemes. The big challenge in the biomedical domain is the high variance of medical "dialects" that exists between different doctors and even stronger between different hospital. The most promising approach so far is based on layers of annotation schemes. The lower the layer, the more reliable are the information. For example numbers (such as *"3"*, *"2.5"* or *"three"*) and explicit dates (such as *"12.10.2003"*) are on the lowest level, while length measurements (such as *"3 cm"* or *"1.3 mm"*) are build on numbers and are a few level above them. We also try to separate more or less standardized information from information that might be influence by "dialects". For example when detecting examinations in combination with location: "CT Thorso: ..." is a much more reliable link than "... Beside the previously mentioned CT we also looked at the thorso ...". We hope that this will render our overall annotation scheme more robust, flexible and maintainable.

7 Conclusion and Future Research

Given its complex nature, general solutions to text understanding are not available yet, however, the business need is here now. This problem can meanwhile only be successfully addressed with specialized approaches, such as rule-based or statistical annotation schemes, to the domain or customer needs. This in return shift the need to enable developers to quickly develop and test these annotation schemes. By speeding up the development process and providing rapid feedback, the developer can focus more time into building and improving the annotation schemes themselves, since there are many possible solutions for a problem, but most of them are imprecise. The quality of the annotations strongly depend on the skills of the developer formulating them.

In the future we would like to take advantage of the build in functionality of custom annotators and research possible topological or statistical methods while having an annotated input.

Acknowledgments. This work was supported by IBM Austria. We are grateful for the support of Niko Marek, Stefan Jaschke and Michael Grosinger.

References

1. Holzinger, A., Geierhofer, R., Modritscher, F., Tatzl, R.: Semantic information in medical information systems: Utilization of text mining techniques to analyze medical diagnoses. Journal of Universal Computer Science 14(22), 3781–3795 (2008)
2. Kreuzthaler, M., Bloice, M., Faulstich, L., Simonic, K., Holzinger, A.: A comparison of different retrieval strategies working on medical free texts. Journal of Universal Computer Science 17(7), 1109–1133 (2011)

3. Gregory, J., Mattison, J.E., Linde, C.: Naming notes - transitions from free-text to structured entry. Methods of Information in Medicine 34(1-2), 57–67 (1995)
4. Holzinger, A., Kainz, A., Gell, G., Brunold, M., Maurer, H.: Interactive computer assisted formulation of retrieval requests for a medical information system using an intelligent tutoring system. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 431–436. AACE, Charlottesville (2000)
5. Lovis, C., Baud, R.H., Planche, P.: Power of expression in the electronic patient record: structured data or narrative text? International Journal of Medical Informatics 58, 101–110 (2000)
6. Blandford, A., Attfield, S.: Interacting with information. Synthesis Lectures on Human-Centered Informatics 3(1), 1–99 (2010)
7. Holzinger, A.: On knowledge discovery and interactive intelligent visualization of biomedical data - Challenges in Human Computer Interaction & Biomedical Informatics (2012)
8. Beale, R.: Supporting serendipity: Using ambient intelligence to augment user exploration for data mining and web browsing. International Journal of Human-Computer Studies 65(5), 421–433 (2007)
9. Funk, P., Xiong, N.: Case-based reasoning and knowledge discovery in medical applications with time series. Computational Intelligence 22(3-4), 238–253 (2006)
10. Holzinger, A., Scherer, R., Seeber, M., Wagner, J., Müller-Putz, G.: Computational Sensemaking on Examples of Knowledge Discovery from Neuroscience Data: Towards Enhancing Stroke Rehabilitation. In: Böhm, C., Khuri, S., Lhotská, L., Renda, M.E. (eds.) ITBAM 2012. LNCS, vol. 7451, pp. 166–168. Springer, Heidelberg (2012)
11. Pinker, S., Bloom, P.: Natural-language and natural-selection. Behavioral and Brain Sciences 13(4), 707–726 (1990)
12. Nowak, M.A., Komarova, N.L., Niyogi, P.: Computational and evolutionary aspects of language. Nature 417(6889), 611–617 (2002)
13. Hauser, M.D., Chomsky, N., Fitch, W.T.: The faculty of language: What is it, who has it, and how did it evolve? Science 298(5598), 1569–1579 (2002)
14. Waldrop, M.M.: Natural-language understanding. Science 224(4647), 372–374 (1984)
15. Weizenbaum, J.: Eliza - a computer program for study of natural language communication between man and machine. Communications of the ACM 9(1), 36–45 (1966)
16. Turing, A.M.: Computing machinery and intelligence. Mind 59(236), 433–460 (1950)
17. Yndurain, E., Bernhardt, D., Campo, C.: Augmenting mobile search engines to leverage context awareness. IEEE Internet Computing 16(2), 17–25 (2012)
18. Erhardt, R.A.A., Schneider, R., Blaschke, C.: Status of text-mining techniques applied to biomedical text. Drug Discovery Today 11(7-8), 315–325 (2006)
19. Lee, W.B., Wang, Y., Wang, W.M., Cheung, C.F.: An unstructured information management system (uims) for emergency management. Expert Systems with Applications 39(17), 12743–12758 (2012)
20. Zins, C.: Conceptual approaches for defining data, information, and knowledge: Research articles. J. Am. Soc. Inf. Sci. Technol. 58(4), 479–493 (2007)
21. Boisot, M., Canals, A.: Data, information and knowledge: have we got it right? IN3 Working Paper Series (4) (2004)
22. Mitkov, R.: The Oxford Handbook of Computational Linguistics (Oxford Handbooks in Linguistics S.). Oxford University Press (2003)

23. Ferrucci, D., Lally, A.: Building an example application with the unstructured information management architecture. IBM Systems Journal 43(3), 455–475 (2004)
24. Nasukawa, T., Nagano, T.: Text analysis and knowledge mining system. IBM Systems Journal 40(4), 967–984 (2001)
25. Gotz, T., Suhre, O.: Design and implementation of the uima common analysis system. IBM Systems Journal 43(3), 476–489 (2004)
26. Mack, R., Mukherjea, S., Soffer, A., Uramoto, N., Brown, E., Coden, A., Cooper, J., Inokuchi, A., Iyer, B., Mass, Y., Matsuzawa, H., Subramaniam, L.V.: Text analytics for life science using the unstructured information management architecture. IBM Systems Journal 43(3), 490–515 (2004)
27. Holzinger, A., Simonic, K., Yildirim, P.: Disease-disease relationships for rheumatic diseases: Web-based biomedical textmining and knowledge discovery to assist medical decision making (2012)
28. Holzinger, A., Yildirim, P., Geier, M., Simonic, K.-M.: Quality-based knowledge discovery from medical text on the Web Example of computational methods in Web intelligence. In: Pasi, G., Bordogna, G., Jain, L.C. (eds.) Qual. Issues in the Management of Web Information. ISRL, vol. 50, pp. 145–158. Springer, Heidelberg (2013)
29. Garvin, J.H., DuVall, S.L., South, B.R., Bray, B.E., Bolton, D., Heavirland, J., Pickard, S., Heidenreich, P., Shen, S.Y., Weir, C., Samore, M., Goldstein, M.K.: Automated extraction of ejection fraction for quality measurement using regular expressions in unstructured information management architecture (uima) for heart failure. Journal of the American Medical Informatics Association 19(5), 859–866 (2012)
30. Clark, A., Fox, C., Lappin, S. (eds.): The Handbook of Computational Linguistics and Natural Language Processing. Blackwell Handbooks in Linguistics. John Wiley & Sons (2010)
31. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge (1999)
32. Schmid, H.: Probabilistic part-of-speech tagging using decision trees (1994)