Klaus P. Jantke
Aran Lunzer
Nicolas Spyratos
Yuzuru Tanaka (Eds.)

LNAI 3847

Federation over the Web

International Workshop
Dagstuhl Castle, Germany, May 2005
Revised Selected Papers

Springer

Klaus P. Jantke Aran Lunzer
Nicolas Spyratos Yuzuru Tanaka (Eds.)

Federation over the Web

International Workshop
Dagstuhl Castle, Germany, May 1-6, 2005
Revised Selected Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Klaus P. Jantke
FIT Leipzig, Forschungsinstitut für InformationsTechnologien
Postfach 30 11 66, 04251 Leipzig, Germany
E-mail: jantke@meme.hokudai.ac.jp

Aran Lunzer
Yuzuru Tanaka
Hokkaido University, Meme Media Laboratory
North 13 West 8, Sapporo 060-8628, Japan
E-mail: {aran,tanaka}@meme.hokudai.ac.jp

Nicolas Spyratos
Université Paris-Sud, Laboratoire de Recherche en Informatique
LRI-Bât. 490, 91405 Orsay Cedex, France
E-mail: spyratos@lri.fr

Library of Congress Control Number: 2005938389

CR Subject Classification (1998): I.2, H.2.8, H.3, H.4, J.1

ISSN 0302-9743
ISBN-10 3-540-31018-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31018-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11605126 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3847

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

The lives of people all around the world, especially in industrialized nations, continue to be changed by the presence and growth of the Internet. Its influence is felt at scales ranging from private lifestyles to national economies, boosting the pace at which modern information and communication technologies influence personal choices along with business processes and scientific endeavors.

In addition to its billions of HTML pages, the Web can now be seen as an open repository of computing resources. These resources provide access to computational services as well as data repositories, through a rapidly growing variety of Web applications and Web services.

However, people's usage of all these resources barely scratches the surface of the possibilities that such richness should offer. One simple reason is that, given the variety of information available and the rate at which it is being extended, it is difficult to keep up with the range of resources relevant to one's interests. Another reason is that resources are offered in a bewildering variety of formats and styles, so that many resources effectively stand in isolation.

This is reminiscent of the challenge of enterprise application integration, familiar to every large organization be it in commerce, academia or government. The challenge arises because of the accumulation of information and communication systems over decades, typically without the technical provision or political will to make them work together. Thus the exchange of data among those systems is difficult and expensive, and the potential synergetic effects of combining them are never realized.

Motivation for overcoming this challenge with respect to the Web is found in many domains. In academia there is a recognized need for interdisciplinary and international availability, distribution and exchange of intellectual resources, which include both static information (such as publications and other research results) and the tools that support research activities. Similar pressure derives from the development and deployment of pervasive computing, which extends the types of resources that are present on the Web to encompass embedded devices and mobile computing resources, communicating over wireless networks. In such domains, just as for enterprise application integration, the rich variety of resources and the boundless human creativity applied in developing new solutions conspire to increase the number of compatibility barriers.

To what extent can this challenge be addressed by standardization? While the development and broad adoption of standards are crucial to the advance of information and communication technologies, any attempt to find a general solution to resource incompatibility through global standardization is doomed to fail because of the diversity of resources and the pace at which they change. That said, given an appropriately narrowed target scope, it is reasonable for providers to agree on a shared middle ground that will increase the mutual

compatibility of their resources. This is the key to the approach known as *mediation*. Mediation, at least as usually understood in the area of databases, requires cooperation among providers in specifying (a) a well-defined community of cooperating sources, and (b) a common schema (called the mediator schema) to which the sources address their queries and/or provide answers.

In contrast to mediation, the study of *federation* involves working to bridge the differences between resources without such a predefined common ground. The process of resource federation in general involves selecting the resources that are to be combined, discovering the relationships that will allow them to work together, establishing the necessary connections, then driving the assembly in a coordinated way to achieve certain desired goals. The science of federation requires new theoretical foundations and enabling technologies for analysis of syntactic and/or semantic interrelations among resources, matching of service requesters and providers, and reliable and secure establishment and coordination of their execution.

Federation over the Web has attracted the attention of researchers aiming to support interdisciplinary and cross-border reuse and interoperation of heterogeneous intellectual resources, for example in support of scientific research, simulation, and digital libraries. Federation over enterprise intranets, also based on Web technologies, is being pursued as a way to bring large numbers of legacy application systems into cooperation with each other.

Existing work on federation can be divided broadly into programmatic and interactive approaches. Programmatic approaches are based on standardization at the level of communication protocols and languages for discovering compatibilities between resources, including the run-time matching of requesters and providers; for the Web, such federation tends to be based on Web-service technologies. On the other hand, interactive federation places in the hands of the user all responsibility for judging which resources are suitable for connection, then establishing and coordinating such connections. A simple example of interactive federation on the Web is the use of visual operations to connect result elements of one Web application, found at predictable locations within its HTML results, to input fields within an HTML form of another application.

From 1 to 6 May 2005 we held a workshop at Dagstuhl Castle to discuss advances in this area, drawing together active researchers from several institutes in Japan and Europe. The workshop focused on theoretical foundations and enabling technologies for federation of resources offered over the Web or within pervasive computing environments. We invited the participants to present and discuss work falling under any of the following topics:

- Knowledge look-up and matching
- Knowledge search and clustering
- Knowledge ontology and mediation
- Interoperation of Web-based resources
- Knowledge extraction and Web wrappers
- Computational models for knowledge federation

Based on the 18 workshop presentations, we went through a process of consultation, reviewing and editing to arrive at the 12 papers in this book. As shown in the table of contents, these papers touch on most of the above topics.

Future research and development of Web-based federation stands to influence how humans use intellectual resources in local and global networks. In combination with the rise of ubiquitous computing, introducing new forms of mobile computing devices and smart objects, computer systems will increasingly form location-based, on-demand federations. By the meeting and cooperation of these systems, humans will be dynamically connected with other humans and with a greater variety of systems and services. One can envisage future workshops on resource federation including contributions from the humanities, including sociology and psychology.

Let the present volume set the stage for such exciting developments.

October 2005

Klaus P. Jantke
Aran Lunzer
Nicolas Spyratos
Yuzuru Tanaka

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005.

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005.

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005.

Vol. 3782: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.), Professional Knowledge Management. XXIII, 739 pages. 2005.

Vol. 3735: A. Hoffmann, H. Motoda, T. Scheffer (Eds.), Discovery Science. XVI, 400 pages. 2005.

Vol. 3734: S. Jain, H.U. Simon, E. Tomita (Eds.), Algorithmic Learning Theory. XII, 490 pages. 2005.

Vol. 3721: A.M. Jorge, L. Torgo, P.B. Brazdil, R. Camacho, J. Gama (Eds.), Knowledge Discovery in Databases: PKDD 2005. XXIII, 719 pages. 2005.

Vol. 3720: J. Gama, R. Camacho, P.B. Brazdil, A.M. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005. XXIII, 769 pages. 2005.

Vol. 3717: B. Gramlich (Ed.), Frontiers of Combining Systems. X, 321 pages. 2005.

Vol. 3702: B. Beckert (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. XIII, 343 pages. 2005.

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005.

Vol. 3690: M. Pěchouček, P. Petta, L.Z. Varga (Eds.), Multi-Agent Systems and Applications IV. XVII, 667 pages. 2005.

Vol. 3684: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part IV. LXXIX, 933 pages. 2005.

Vol. 3683: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LXXX, 1397 pages. 2005.

Vol. 3682: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LXXIX, 1371 pages. 2005.

Vol. 3681: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXXX, 1319 pages. 2005.

Vol. 3673: S. Bandini, S. Manzoni (Eds.), AI*IA 2005: Advances in Artificial Intelligence. XIV, 614 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005.

Vol. 3661: T. Panayiotopoulos, J. Gratch, R.S. Aylett, D. Ballin, P. Olivier, T. Rist (Eds.), Intelligent Virtual Agents. XIII, 506 pages. 2005.

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 pages. 2005.

Vol. 3651: R. Dale, K.-F. Wong, J. Su, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2005. XXI, 1031 pages. 2005.

Vol. 3642: D. Ślęzak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIII, 738 pages. 2005.

Vol. 3641: D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005.

Vol. 3635: J.R. Winkler, M. Niranjan, N.D. Lawrence (Eds.), Deterministic and Statistical Methods in Machine Learning. VIII, 341 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005.

Vol. 3630: M.S. Capcarrère, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (Eds.), Advances in Artificial Life. XIX, 949 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005.

Vol. 3620: H. Muñoz-Ávila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005.

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005.

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3601: G. Moro, S. Bergamaschi, K. Aberer (Eds.), Agents and Peer-to-Peer Computing. XII, 245 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E.T. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A.K. Dey, B. Kokinov, D.B. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3550: T. Eymann, F. Klügl, W. Lamersdorf, M. Klusch, M.N. Huhns (Eds.), Multiagent System Technologies. XI, 246 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrović (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.-B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V.A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E.P. Stabler, J.V. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3490: L. Bolc, Z. Michalewicz, T. Nishida (Eds.), Intelligent Media Technology for Communicative Intelligence. X, 259 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3487: J.A. Leite, P. Torroni (Eds.), Computational Logic in Multi-Agent Systems. XII, 281 pages. 2005.

Vol. 3476: J.A. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World V. XIII, 349 pages. 2005.

Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), Massively Multi-Agent Systems I. XI, 349 pages. 2005.

Vol. 3445: G. Chollet, A. Esposito, M. Faúndez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications. XIII, 433 pages. 2005.

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 pages. 2005.

Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), Active Mining. XII, 349 pages. 2005.

Vol. 3419: B.V. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. X, 217 pages. 2005.

Vol. 3416: M.H. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3413: K. Fischer, M. Florian, T. Malsch (Eds.), Socionics. X, 315 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F.P. M. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems II. VIII, 313 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005.

Vol. 3374: D. Weyns, H. V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Table of Contents

Knowledge Evolution

Text Mining Using Markov Chains of Variable Length

Björn Hoffmeister[1] and Thomas Zeugmann[2]

[1] RWTH Aachen, Lehrstuhl für Informatik VI, Ahornstr. 55, 52056 Aachen
hoffmeister@i6.informatik.rwth-aachen.de
[2] Division of Computer Science, Hokkaido University,
N-14, W-9, Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

Abstract. When dealing with knowledge federation over text documents one has to figure out whether or not documents are related by context. A new approach is proposed to solve this problem.

This leads to the design of a new search engine for literature research and related problems. The idea is that one has already some documents of interest. These documents are taken as input. Then all documents known to a classical search engine are ranked according to their relevance. For achieving this goal we use Markov chains of variable length.

The algorithms developed have been implemented and testing over the Reuters-21578 data set has been performed.

1 Introduction

When one is aiming at knowledge federation over the web, one is often looking for information around a specific topic. In a first step, one may find one or more papers dealing with the topic of interest. Then, the next task is to find related papers. Another situation to which our research may apply is to enable documents to communicate to one another when trying to form a knowledge federation over the web. Again, in such cases it may be very important to answer a question like "is document A on the same subject as document B?" If the answer is affirmative, then a federation is made, otherwise it is rejected.

For dealing with such problems, we propose an approach based on Markov Chains of variable length. We exemplify this approach by constructing a search engine taking as inputs papers and returning a list of semantically related papers.

Currently used search engines do not take documents as input. They rely on queries of one or a few words describing the desired information. Basically, there are two different search strategies.

The first concept is based on catalogues. A catalogue contains similar objects, e.g., web-sites about machine learning. Hence, a query to such a catalogue system is answered with a certain set of catalogues. Each of them ideally carries objects relevant to the query. Search engines in libraries and web directories like Yahoo![1]

[1] http://www.yahoo.com

K.P. Jantke et al. (Eds.): Federation over the Web, LNAI 3847, pp. 1–24, 2006.

are based on this approach. The quality depends on the quality of the catalogues. Producing good catalogues is still time consuming and expensive.

The second strategy is to perform a full-text search over all available documents. Common web search engines like Google[2] and AltaVista[3] are based on this concept. The disadvantage of a full-text search is the large number of matches. Therefore, a ranking is introduced and only the top ranked documents are returned. Google's main ranking criterion is the linkage rate of a web-site, that is, the more pages link to the document or web-site the higher the rank.

AltaVista uses a syntactical concept. It ranks the results depending on criteria like the positions of and distances between the queried words in the document. So, the alignment of the words should reflect the relevance of the document.

Both strategies have their advantages and disadvantages. Moreover, both approaches fail, for example, if the query allows ambiguities (cf. [13]). And the ranking criteria may overlook relevant documents or give them a low ranking, since simple queries do not allow a fine-grained ranking of relevance.

Now, the idea is to combine the advantages of both approaches. Our search engine takes a set of documents as query, classifies them, and ranks all the documents known by the search engine according to their relevance. To receive a ranking based on semantical relevance we use a model, which can keep more of the meaning of a document than common *data representation models.*

Following Ron *et al.* [18], we tried to use the *variable memory Markov model* defined as a *prediction suffix tree* (abbr. PST). So, we arrive at a *Markov model with variable memory*, or *n-gram VMM model* for short which is used for text representation. The *n-gram VMM model* is learned by statistical inference, a special form of inductive learning. Then we combine text retrieval and text classification.

We shortly outline the underlying mathematical background, describe the workflow of the resulting search engine, and report experimental results.

2 Preliminaries

Natural language is the most common form to exchange information between human beings, e.g., news stories are published in natural language as well as scientific papers. These documents often contain additional information encoded in structured text, like tables or formulas, or in graphical form. However, we shall only use the text in a document. Such a reduction may waste information. But for the particular setting we study within this paper, i.e., the Reuters Data set, it is sufficient. Additionally, all documents in this data set are written in English. Therefore, we restrict ourselves to deal with English texts.

We assume familiarity with formal language theory (cf., e.g., [9]). The word is used as smallest unit. In the literature, one also finds many other possible atomic units. Research has been done using sub-word units like letters or morphemes on the one hand and multi-word units, i.e., combinations of one or more words, on the other hand, e.g., see [13], [10], and [19].

[2] http://www.google.com
[3] http://www.av.com

We continue with technical notations. $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of all natural numbers, and $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. By Σ we denote a fixed finite alphabet, Σ^* denotes the free monoid over Σ, and $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$, where ϵ is the empty word. An *n-gram* is a string of $n \in \mathbb{N}^+$ concatenated letters. The set of all *n-grams* over Σ is denoted by Σ^n, where $\Sigma^0 = \{\epsilon\}$. We use $\Sigma^{\leq n}$ to denote $\bigcup_{i=1}^{n} \Sigma^i$.

Our alphabet is the set of all English words, i.e., a suitable subset of the English vocabulary which we denote by $\mathbb{V}$. Thereby we have to assure that $\mathbb{V}$ is a set of indivisible symbols such that there exist an one-to-one mapping between the symbols in $\mathbb{V}$ and the words in the English dictionary. The words of the vocabulary are written in another alphabet which we denote by $\mathbb{A}$. The relation between a word symbol in $\mathbb{V}$ and its representation in $\mathbb{A}^+$ is expressed by a mapping $\omega : \mathbb{V} \to \mathbb{A}^+$, where $\omega(\cdot)$ is injective. This can be easily achieved by introducing a delimiter symbol β such that $\beta \notin \mathbb{A}$.

Note that we use the term *word* to refer to member of $\mathbb{V}$. Therefore, an *n-gram* $s = \sigma_1 \ldots \sigma_n$, $\sigma_i \in \mathbb{V}$, $1 \leq i \leq n$, is a concatenation of n words and a string refers to any *n-gram*, $n \in \mathbb{N}$. A phrase is a meaningful concatenation of two or more words; technically any *n-gram*, $n > 1$, occurring in a document is a phrase. And finally, a term is either a word or a phrase. A document is then a sequence of sentences, where a sentence is a concatenation of words from $\mathbb{V}$.

For dealing with document classification and retrieval we use probabilistic language models. The idea is that documents dealing with different subjects also use a different subset of the vocabulary $\mathbb{V}$ and even different phrases over these subsets. For example, a document about stock exchange might contain words like "hausse" and "baisse", which will almost never appear in a text about machine learning. So, the observation is that texts about different subjects differ in the used words. Furthermore, terms like "machine learning" or "conditional mutual information" are surely not part of texts about stock markets, but the single words "machine", "learning", "conditional", "mutual", and "information" may occur in such a text. Moreover, the idea is to look at how likely a word is, if the previous words are known. In a text about machine learning it is very likely that "machine" is followed by "learning", where in a text about stock market exchange it is probably followed by "manufacture" or "supplier", but not by "learning".

The task of predicting the next word given the previous words is called *language modeling task* and a model solving the task is called a *generative model*, see [13] and [8]. Therefore, we continue with the following definitions.

Definition 1 (Stochastic model). *A stochastic model or process is a sequence of random variables* $(X_t)_{t \in \mathbb{N}}$.

Let us assume every random variable in $(X_t)_{t \in \mathbb{N}}$ has the same range $\mathcal{X}$. Thus, the statistical properties of $(X_t)_{t \in \mathbb{N}}$ are completely determined by the nth-order probability distribution $p(x_0, x_1, \ldots, x_n) := P(X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n)$, $x_i \in \mathcal{X}$, $0 \leq i \leq n$, $n \in \mathbb{N}$, see [16].

Moreover, we use $\mathcal{L}$ to denote the language used by subject S, i.e., $\mathcal{L} \subset \mathbb{V}^*$. We then expect two documents to be about the same subject and hence semantically related, if the subjects of the documents use the same language. But we shall use probabilities instead of absolute statements. That is, we do not wish to decide

whether or not a string or a sentence is in $\mathcal{L}$. Instead the language model we are aiming at returns for every string $s \in \mathbb{V}^*$ the probability for s to be in $\mathcal{L}$.

Let S be a subject, let $\mathcal{L}$ be the language of S, and let p_S be the probability distribution underlying $\mathcal{L}$. Furthermore, let M be a *generative model* for S. Thus, M solves the *language modeling task* for S, if $p_S(\sigma|s) = p_M(\sigma|s)$ for every $\sigma \in \mathbb{V}$ and for every $s \in \mathbb{V}^*$, where s is the sequence of all preceding words. Obviously, if M solves the *language modeling task* for S, the strings generated by M are distributed according to p_S and hence M is a probabilistic language model for S.

How many of the previous words are necessary for making a good prediction for the next word? The surprising answer is: most often only a few. For example, if we see the word "machine" in a text about machine learning, "learning" is very likely to be the next word, and knowing the words previous to "machine" does not provide much additional information about the likeliness. Manning *et al.* [13] claim that it takes quite a big effort to beat a *generative model* for natural language, which predicts the next word on the previous two words.

In general, good estimations for the next word in natural language are context dependent. An example is provided by this text. As mentioned before, the word "machine" is very likely to be followed by "learning"; but what about "Markov"? In the following, the words "model" and "chain" occurs after "Markov", but the 3-*gram* "variable memory Markov" is always followed by "model". Hence, we want a model, which can capture this property of natural language.

The model which has the desired properties, is an *n-gram Markov model with variable memory*, *n-gram VMM model* for short, which is defined by a *variable memory Markov model*, *VMM model* for short. A *VMM model* in turn is a special kind of the well-known *Markov model*. So, first *Markov models* are shortly repeated, followed by the definition of the *VMM model*, from which we derive the *n-gram VMM model*. In addition, the classical *n-gram Markov model* is presented and compared to our model, which proves to be superior.

If we regard *generative models* as stochastic processes, any random variable of the process has the property of only depending on the previous variables. A special kind of those dependencies is captured by the *Markov model*, where a random variable depends only on its direct predecessor. We shall see that, despite this restriction, the *Markov model* is a suitable base for a *generative model* for a language. In terms of a *Markov model* we call the value of a random variable a *state* and its range *state space*.

Definition 2 (Markov model). *Let $(X_t)_{t\in\mathbb{N}}$ be a stochastic model and let $\mathcal{X}$ be the state space for all random variables $X_t, t \in \mathbb{N}$. $(X_t)_{t\in\mathbb{N}}$ is a Markov model, iff it meets the Markov assumption*

$$P(X_{t+1} = x_{t+1}|X_0 = x_0, \ldots, X_t = x_t) = P(X_{t+1} = x_{t+1}|X_t = x_t) \ . \quad (1)$$

Let $p(\cdot|\cdot)$ be a function $p : \mathcal{X} \times \mathcal{X} \to [0,1]$. The Markov model $(X_t)_{t\in\mathbb{N}}$ is homogeneous, iff it fulfills the time invariance assumption

$$P(X_{t+1} = x_{t+1}|X_t = x_t) = p(x_{t+1}|x_t), \quad \textit{for every } t \in \mathbb{N}. \quad (2)$$

p is the Markov core, where $p(x|y) \geq 0$ and $\sum_{x\in\mathcal{X}} p(x|y) = 1$ for all $x, y \in \mathcal{X}$.

If a random variable depends only on its predecessor, the question remains of how to predict the *state* of the initial random variable X_0. This is done by a special initial distribution. A *Markov model* together with an initial distribution for X_0 leads to the definition of a *Markov chain*. We follow the definition given by [4], because it fits our purpose best. Other definitions do not restrict *Markov chains* to be homogeneous, e.g., see [2].

Definition 3 (Markov chain). *A Markov chain is a homogeneous Markov model $(X_t)_{t\in\mathbb{N}}$ with state space $\mathcal{X}$, Markov core p and initial probability distribution π, where X_0 is distributed according to π.*

Before we continue with the definition of the *variable memory Markov model*, we use the *Markov chain* to define a first probabilistic language model, the classical *n-gram Markov model*. It shows how to use a *Markov model* to derive a language model; the *n-gram VMM model* will be defined analogously. We shall also use it to point to the advantages of our model.

The *n-gram Markov model* is a *generative model* predicting the next word in dependence on the previous n words. Since a *Markov chain*, by its definition, predicts the value of a random variable only on the value of its direct predecessor the following construction is necessary which uses overlapping random variables.

Let $(\Sigma_t)_{t\in\mathbb{N}}$ be a sequence of random variables, where each random variable in $(\Sigma_t)_{t\in\mathbb{N}}$ has range $\mathbb{V}$. We define a second sequence of random variables $(S_t)_{t\in\mathbb{N}}$, where each random variable in $(S_t)_{t\in\mathbb{N}}$ has range $\mathbb{V}^n$, $n \in \mathbb{N}^+$. The relation between $(\Sigma_t)_{t\in\mathbb{N}}$ and $(S_t)_{t\in\mathbb{N}}$ is given by the definition of the following equivalence. Let s be an *n-gram* and let $s = \sigma_0\sigma_1 \ldots \sigma_{n-1}$, $\sigma_i \in \mathbb{V}$, $0 \leq i < n$. Then,

$$S_t = s \quad \stackrel{def}{\Longleftrightarrow} \quad \Sigma_t = \sigma_0, \Sigma_{t+1} = \sigma_1, \ldots, \Sigma_{t+n-1} = \sigma_{n-1}, \tag{3}$$

for every $t \in \mathbb{N}$. Thus, the random variables S_t overlap, i.e., S_t depends on its predecessors, where the dependency is completely described by the direct predecessor S_{t-1}, $t \in \mathbb{N}^+$.

S_t contains the information about n words and hence, for predicting the value of Σ_{t+n} given the previous n words, the knowledge of the value of S_t is sufficient. We express the probability of the value of Σ_{t+n} in terms of S_t and S_{t+1} as follows. Let $S_t = s_0$, let $S_{t+1} = s_1$, and let $s_0 = \sigma_0\sigma_1 \ldots \sigma_{n-1}$, where s_0 and s_1 in $\mathbb{V}^n$, $\sigma_i \in \mathbb{V}$, $0 \leq i < n$. From (3) it follows that $s_1 = \sigma_1 \ldots \sigma_{n-1}\sigma_n$, $\sigma_n \in \mathbb{V}$, and hence

$$\begin{aligned}
&P(S_{t+1} = s_1 |\, S_t = s_0) \\
=&P(\Sigma_{t+1} = \sigma_1, \ldots, \Sigma_{t+n-1} = \sigma_{n-1}, \Sigma_{t+n} = \sigma_n \\
&\qquad |\Sigma_t = \sigma_0, \Sigma_{t+1} = \sigma_1, \ldots, \Sigma_{t+n-1} = \sigma_{n-1}) \\
=&P(\Sigma_{t+n} = \sigma_n |\, \Sigma_t = \sigma_0, \Sigma_{t+1} = \sigma_1, \ldots, \Sigma_{t+n-1} = \sigma_{n-1}) \\
=&P(\Sigma_{t+n} = \sigma_n |\, S_t = s_0), \qquad \text{for every } t \in \mathbb{N}\ .
\end{aligned} \tag{4}$$

Obviously, $(S_t)_{t\in\mathbb{N}}$ fulfills the *Markov assumption* and thus we see how a *Markov chain* can be used to predict a word in dependence on the previous n words. Thus, we arrive at the following definition.

Definition 4 (n-gram Markov model). *Let S denote a subject and let p_S be the probability distribution of the language of S. Furthermore, let $(S_t)_{t\in\mathbb{N}}$ be a Markov chain with state space $\mathbb{V}^n$, Markov core p and initial probability distribution π. $(S_t)_{t\in\mathbb{N}}$ is called n-gram Markov model for S, iff*

$$p_S(\sigma_0\sigma_1\ldots\sigma_{m-1}) = \pi(s_0)p(s_1|s_0)p(s_2|s_1)\ldots p(s_{m-n}|s_{m-n-1}), \tag{5}$$

where $s_i \in \mathbb{V}^n$, $s_i = \sigma_i\sigma_{i+1}\ldots\sigma_{i+n-1}$, $0 \leq i \leq m-n$, for all m-grams $\sigma_0\sigma_1\ldots\sigma_{m-1} \in \mathbb{V}^m$, $m \in \mathbb{N}$, $m \geq n$.

Because p_S is to fulfill Kolmogorov's consistency condition the initial probability distribution π must have the following property, see [1].

Let $s = \sigma_1\sigma_2\ldots\sigma_n$, $\sigma_i \in \mathbb{V}$, $1 \leq i \leq n$, be an *n-gram*. Furthermore, let $\mathrm{suff}(s)$ denote the longest proper suffix of s, i.e., $\mathrm{suff}(s) = \sigma_2\ldots\sigma_n$. Then π must fulfill the equation

$$\pi(\mathrm{suff}(s)\,\sigma) = \sum_{\sigma'\in\mathbb{V}} p(\sigma|\sigma'\mathrm{suff}(s))\;\pi(\sigma'\mathrm{suff}(s)),$$

for every $s \in \mathbb{V}^n$, where $\sigma \in \mathbb{V}$. We get the desired property, if we define $\pi(\sigma_1\ldots\sigma_n)$ as $P(S_1 = s)$, where $s = \sigma_1\ldots\sigma_n$ for all *n-grams* $\sigma_1\ldots\sigma_n \in \mathbb{V}^n$.

Now, we have a first probabilistic language model. But the size of the *state space* is by definition $|\mathbb{V}|^n$. This will lead to problems if $n \geq 2$ when one wants to learn such a model and the documents are too short (cf., e.g., [2]). For seeing the problem, note that a normal vocabulary of natural language has a size of more than 20.000 words. So, in order to estimate all probabilities described above for an *2-gram Markov model* one needs a sample of more than $20.000^3 = 8 \times 10^{12}$ words. Obviously, we normally do not possess such a large sample.

Therefore, we want to use the *variable memory Markov model* which has been defined in a different context by Ron *et al.* [18]. A variable memory Markov model is defined as a *prediction suffix tree*(PST).

Definition 5 (suffix tree). *Let Σ be an alphabet, let $\mathcal{T}$ be a tree and let $\mathbf{E}$ denote the set of edges between the nodes in $\mathcal{T}$. Furthermore, let each edge be labeled by a symbol $\sigma \in \Sigma$ and each node by a string $s \in \Sigma^*$. The two functions $l_{\mathbf{E}} : \mathbf{E} \to \Sigma$ and $l_{\mathcal{T}} : \mathcal{T} \to \Sigma^*$ return the label of an edge and of a node, respectively. $\mathcal{T}$ is a suffix tree over Σ, iff it has the following properties:*

i) *$\mathcal{T}$ has degree $|\Sigma|$.*
ii) *The root node n_0 of $\mathcal{T}$ has label ϵ.*
iii) *For every node $n_l \in \mathcal{T}$, $l \in \mathbb{N}$, and $n_0 \to n_1 \to \ldots \to n_{l-1} \to n_l$, the walk from the root node n_0 to node n_l, the label of n_l equals the concatenated labels of the passed edges, i.e., $l_{\mathcal{T}}(n_l) = l_{\mathbf{E}}(e_{0,1})\,l_{\mathbf{E}}(e_{1,2})\;\ldots\;l_{\mathbf{E}}(e_{l-1,l})$*
iv) *Neither two edges of one node nor two nodes have the same label.*

Definition 6 (next symbol probability function). *Let Σ be an alphabet and let $\gamma_s, s \in \Sigma^*$, be a function. The function γ_s is called next symbol probability function over Σ, iff it defines a probability distribution over Σ.*