3rd International Workshop on Text-Based Information Retrieval (TIR-06)

Intelligent technologies for information mining and retrieval have become an important and exciting field of research in our information-flooded society. Methods of text-based information retrieval receive special attention, which results from the fundamental role of written text, but also because of the high availability of the Internet.

There are various techniques and methods being used for text-based information retrieval tasks, which stem from different research areas: machine learning, computer linguistics and psychology, user interaction and modeling, information visualization, or distributed systems. The development of powerful retrieval tools requires the combination of these developments, and in this sense the workshop shall provide a platform that spans different views and approaches.

The workshop addresses researchers, users, and practitioners from different fields: data mining, machine learning, document and knowledge management, semantic technologies, and information retrieval in general.

Benno Stein
Odej Kao

August 2006

Organization

Program Chairs

Benno Stein, Bauhaus University Weimar, Germany
Odej Kao, University of Paderborn, Germany

Program Committee

Markus Ackermann, University of Leipzig
Mikhail Alexandrov, National Polytechnic Institute of Mexico
Elizabeth Sugar Boese, Colorado State University
Michael Granitzer, Know-Center Graz
Heiko Holzheuer, Lycos Europe
Andreas Hotho, University of Kassel
Wolfgang Kienreich, Know-Center Graz
Theodor Lettmann, University of Paderborn
Mathias Lux, Technical University Graz
Sven Meyer zu Eissen, Bauhaus University Weimar
Oliver Niggemann, DSpace
Henrik Nottelmann, University of Duisburg

Table of Contents

Classifying Encounter Notes in the Primary Care Patient Record

Thomas Brox Røst and **Øystein Nytrø** and **Anders Grimsmo**[1]

Abstract. The ability to automate the assignment of primary care medical diagnoses from free-text holds many interesting possibilities. We have collected a dataset of free-text clinical encounter notes and their corresponding manually coded diagnoses and used it to built a document classifier. Classifying a test set of 2,000 random encounter notes yielded a coding accuracy rate of 49.7 %. Automated coding of primary care encounter notes is a novel application area, and though imperfect our method proves interesting enough to warrant further research.

1 Introduction

In this study we attempt to classify primary care clinical encounter notes into their corresponding diagnoses. We do so by learning document classifiers from a manually coded dataset collected from a Norwegian primary care center. Research have shown that the manual diagnosis coding of primary care encounter notes tend to be of high quality [20]. This, coupled with the the size of the dataset, makes the application area interesting from an information retrieval and document classification point of view. In the long term, being able to infer diagnoses from written text might prove useful in e.g. detection of incorrect diagnoses and improving electronic patient record systems. We consider this study as an initial exploration into applying proven document classification techniques onto a novel application area.

The electronic patient record (EPR) has gradually attained widespread usage in primary care. In Norway, more than 90 % of primary care physicians are routinely using computer-based patient-record systems [3] and many have been doing so for more than 15 years. A typical feature of most commercial EPR systems in use today is that the encounter note, which is the main documentation of the doctor-patient consultation, is written as free-text narrative. There are perfectly practical reasons for this: Unstructured free-text is easy to write and represents the traditional way of documenting patient treatment. However, this makes the information within less suitable for automated processing and thereby keeps the EPR from fulfilling its full potential as a useful tool for both research and clinical practice. Attempts have been made to create EPRs that impose varying degrees of structure on the clinical narrative, but with limited success so far.

To alleviate this problem, many researchers have attempted to use natural language processing (NLP), text classification and text mining techniques on clinical narrative. Some NLP systems have proven very useful in a number of clearly defined domains, such as detection of bacterial pneumonia from chest X-ray reports [4], finding adverse drug events in outpatient medical records [10] and discharge summaries [19], and identifying suspicious findings in mammogram reports [12]. A common feature of such systems is that they restrict themselves to a narrow clinical domain with a clearly defined vocabulary and a limited form of discourse, such as one would find in specialized hospital reports. Our long-term goal is to draw on research from these areas and explore the usefulness of similar techniques on the primary care patient record. However, the lack of empirical knowledge on the content in primary care documentation raises the need for preliminary investigations on the narrative structure found therein. This initial study attempts to use supervised document classfication to explore if there is a correspondence between the diagnosis and the documented encounter. Besides from the previously mentioned possible benefits of automated coding, a secondary purpose is to learn more about the informational value and underlying documentational patterns in primary care encounter notes.

2 Background

Among the characteristic features of primary care encounter notes are sparseness, brevity, heavy use of abbreviations and many spelling mistakes. The notes are normally written during the consultation by the treating physician, this in contrast with hospital patient records which are usually dictated by the physician and then transcribed by a secretary. A typical encounter note might look something like this:

> Inflamed wounds over the entire body. Was treated w/ apocillin and fucidin cream 1 mth. ago. Still using fucidin. Taking sample for bact. Beginning tmnt. with bactroban. Call in 1 week for test results[2].

To classify such notes we rely on the presence of manually coded diagnosis codes. The use of clinical codes in primary care is common in the United Kingdom, the Netherlands, and Norway [16]. The motivation for coding is both for reimbursement and statistical purposes. In our experimental dataset the notes are coded according to the ICPC-2 coding system. ICPC-2 is the second edition of the International Classification of Primary Care, a coding system which purpose is to provide a classification that reflects the particular needs and aspects of primary care [11]. Using a single ICPC code, each health care encounter can be classified so that both the reasons for encounter, diagnoses or problems, and process of care are evident. Together, these elements make out the core constituent parts of the health care encounter in primary care. Moreover, one or more encounters associated with the same health problem or disease form an episode of care [9].

[1] Department of Computer and Information Science and The Norwegian EHR Research Centre, Norwegian University of Science and Technology, Trondheim, Norway. Emails: {brox, nytroe}@idi.ntnu.no, anders.grimsmo@medisin.ntnu.no

[2] Translated from the Norwegian

ICPC-2 follows a bi-axial structure with 17 chapters along one axis and 7 components along the other. The chapters are single-letter representations of body systems (Table 1) while the components are two-digit numeric values (Table 2). As an example, "R02" is the ICPC code for shortness of breath.

Table 1. ICPC chapter codes.

Chapter code	Description
A	General and unspecified
B	Blood, blood-forming organs and immune mechanism
D	Digestive
F	Eye
H	Ear
K	Circulatory
L	Musculoskeletal
N	Neurological
P	Psychological
R	Respiratory
S	Skin
T	Endocrine, metabolic and nutritional
U	Urological
W	Pregnancy, child-bearing, family planning
X	Female genital
Y	Male genital
Z	Social problems

Table 2. ICPC component codes.

Number	Range	Description
1	01-29	Complaint and symptom component
2	30-49	Diagnostic, screening, and preventive component
3	50-59	Medication, treatment, procedures component
4	60-61	Test results component
5	62-63	Administrative component
6	64-69	Referrals and other reasons for encounter
7	70-99	Diagnosis/disease component

There are several examples of attempts to automate the coding of diagnoses [5, 15, 18, 21, 23], all of which concern themselves with the alternative ICD code. ICD is a more complex code than ICPC and is more suited for specialized usage in hospitals. March [18] describes the use of Bayesian learning to achieve automated ICD coding of discharge diagnoses. Franz [5] compares coding methods with and without the use of an underlying lexicon and concludes that lexicon-based methods perform no better than lexicon-free methods, unless one adds conceptual knowledge. Larkey [15] found that using a combination of different classifiers yielded improved automatic assignment of ICD codes. There is a practical purpose to automated ICD coding: ICD is a more complex code than ICPC and accordingly manual ICD encoding takes up a lot of time. There have also been other approaches towards automated coding of clinical text. Hersh [8] attempted to predict trauma registry procedure codes from emergency room dictations. Aronow [2] classified encounter notes in order to find acute exacerbations of asthma and radiology reports for certain findings, this through the use of Bayesian inference networks and the ID3 decision tree algorithm. Document classification and IR has been applied in other medical domains as well, such as clustering of medical paper abstracts [17].

Examples of automated ICPC coding are harder to come by. Letrilliart [16] describes a string matching system that assigns ICPC codes from free-text sentences containing hospital referral reasons, based on a manually created look-up table. We have not found examples of similar attempts at automated ICPC classification in the literature.

As for classification techniques, this study uses support vector machines (SVM). SVMs have proved useful and have shown good general performance for text classification tasks [13] when compared with other classifiers. Our goal for this study is not to compare classification methods; this will be explored further in future work.

3 Methods and Data

We have collected a dataset from a medium-sized general practice office in Norway. The data consists of encounter notes for a total of 10,859 patients in the period from 1992 to 2004. All in all, there are 482,902 unique encounters. The Norwegian Health Personnel Act [1] requires that caregivers provide "relevant and necessary information about the patient and about the health care" in the patient record. In practice, this manifests itself as a combination of structured and unstructured information about the encounter. Information such as personal details about the patient, prescriptions, laboratory results, medical certificates and diagnosis codes is typically available in structured format, while encounter notes, referrals and discharge notes comes in the form of unstructured free-text. For the purposes of this paper, we have only considered the encounter notes and the accompanying ICPC-2 diagnosis code.

A known source of noise is that a minority of the notes are likely to be written in Danish or *nynorsk* (literally "New Norwegian") rather than standard Norwegian (*bokmål*). There are also more than 20 different authors, so there may be differences in documentational style as well. Interns fresh out of medical school may for example be more inclined to document more thoroughly than an experienced physician.

The dataset has been automatically anonymized using a custom-built anonymization tool [22]. Each word or token is controlled against a database of words that are known to be insensitive and a set of rules that deal with alphanumeric patterns such as medication doses, date ranges, and laboratory test values. Sensitive tokens are replaced with a general identifier or an identifier that shows the type of token that was replaced.

Each encounter will typically consist of a written note of highly variable length and zero or more accompanying ICPC codes. 287,868 of the available encounters have one or more ICPC codes (Table 3).

Table 3. Number of ICPC codes per encounter.

Number of ICPC codes	Number of encounters
1	235,860
2	44,651
3	6,037
$\geq$4	1,320

There are some notable differences in terms of code use between hospital and primary care settings. Larkey [15] describes a test set of discharge summaries with a mean of 4.43 ICD-9 codes per document, while Nilsson [20] notes that a set of Swedish general practice patient records has a mean of 1.1 ICD-10 codes per record. While there may be regional and cultural differences with respect to coding practice, the latter corresponds with our findings of 1.2 ICPC-2 codes per note (Table 3).

Since we concern ourselves with the relation between the encounter note and the ICPC code, we discard all encounters with more

than one code in order to avoid ambiguity in the training data. Of the 235,860 encounters that are left, 175,167 have an accompanying encounter note.

The use of ICPC codes as classification bins for encounter notes is essentially a multi-class classification problem. Since there are 726 distinct ICPC codes it becomes practical to reduce the class dimensionality. We choose to group codes according to their chapter value, so that we are left with the 17 single-letter body codes as classes.

When grouping encounter notes by their ICPC chapter value we note that there is a varying degree of verbosity. The use of sparse encounter notes is often common in primary care, for instance when renewing recurring prescriptions. To determine average note verbosity for each ICPC chapter, all relevant encounter notes are tokenized. After removing stop words, whitespace and other noisy elements, the average length and standard deviation is calculated (Table 4).

Table 4. Average note length by ICPC chapter.

Chapter	Avg. No words	St. dev.	Samples
N (Neurological)	40	33.2	5,637
D (Digestive)	39	30.0	11,386
Z (Social)	36	35.1	570
X (Female genital)	36	27.1	6,244
P (Psychological)	32	35.6	9,939
A (General)	32	28.9	12,052
Y (Male genital)	31	24.9	1,993
F (Eye)	31	23.5	4,998
L (Musculoskeletal)	29	26.8	36,493
R (Respiratory)	28	21.8	22,846
K (Circulatory)	27	25.6	21,089
H (Ear)	27	21.3	5,526
W (Pregnancy)	26	24.5	5,614
U (Urological)	26	25.2	4,502
T (Endocrine)	26	22.4	5,498
S (Skin)	26	20.3	18,432
N/A	23	20.6	6,545
B (Blood)	22	23.3	2,348

We note that Larkey's discharge summaries [15] has a mean length of 633 words, which is more than an order magnitude higher than our notes. Notwithstanding cultural and institutional differences, this highlights how hospital discharge summaries usually provide a more self-contained description of the patient and his ailments. In the Norwegian health care system the patient will typically use just one primary care physician who acts as a gatekeeper for specialized hospital care when necessary. Accordingly descriptions of the patient's state may span several encounter notes in the primary care patient record.

Since many classification techniques, including support vector machines (SVM), are restricted to dealing with binary classification tasks, we have to reduce our multi-class classification task into a set of binary tasks. For each pair of classes $(i, j) : i, j \in \{A, B, \ldots, Z\}$ where $i, j = 1 \ldots c, j \neq i$ we create a two-class classifier $< i, j >$. If c is the number of classes, we end up with $c(c-1)$ binary classifiers, or $17 \times 16 = 272$ in this case. This technique is known as double round robin classification [6]. The classifier $< i, j >$ will then solely consist of training examples from encounter notes with ICPC chapter codes i and j. To determine the final predicted class of any given note we feed it through each classifier and record the result. The class that receives the highest number of predictions is chosen to be the most likely one. In case of ties we choose the class with the highest number of occurrences in the training set, or, as a last resort, pick one at random. To build and run the classifiers we used the SVM-Light[3] toolkit.

We use word and phrase frequencies as the base component when constructing feature vectors for the classifiers. If we were to rely on single words alone we would lose some contextual information [8], so frequency counts are performed on all unigrams, bigrams and trigrams in the encounter note, excluding stop words. The occurrence of an n-gram is recorded as a *true* value in the feature vector. While n-grams may be a simplistic way of representing context, it still allows us to catch phrases and turns of words that may have discerning qualities.

As is common with word-based feature vectors, it is useful to apply some dimension-reducing technique to limit the size of the vector. The challenge lies in pruning those features that are the most inconsequential to the classifier's predictive qualities. For this experiment we adapt a technique described in [14]. For each classifier the frequency of all unigrams, bigrams and trigrams occurring in all training notes for both classes are counted. If an n-gram occurs in more than 7.5 % of either the true or the false class notes it is tagged as a likely candidate for inclusion. All candidates are then ranked according to their true class frequency to false class frequency ratio. Finally the top 100 candidates are chosen as the most relevant features. As an example, Table 5 shows the 20 first selected features from the F (Eye) versus P (Psychological) classifier.

Table 5. F versus P classifier, 20 most relevant features.

Original n-gram	Appr. English translation	Comment
kloramf	chloramph	Abbreviation
cornea	cornea	
øyelokk	eyelid	
rusk	dust	
hø øye	right eye	Abbreviation
kloramfenikol	chloramphenicol	
rdt	red	
ve øye	left eye	Abbreviation
øye	eye	
øyet	the eye	
injeksjon	injection	
puss	pus	
øyne	eyes	
hø	right	Abbreviation
ve	left	Abbreviation
begge	both	Abbreviation
ved us	after examination	Abbreviation
us	examination	Abbreviation
lett	easily	
ser	sees	

2.000 notes were selected at random from the 175.167 available notes to be used as a test set; the remaining notes were used to train the classifiers. As seen from Table 4, this implies that the amount of training data available for each classifier will differ.

4 Results

Table 6 shows the results from attempting to classify the 2.000 test cases. A total of 994 cases were classified correctly, giving an overall accuracy rate of 49.7 %. As a comparison, guessing for the most frequent chapter code (L) all the time will yield an accuracy of 20.8 %. The displayed results are from a single test run.

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

Table 6. Predicted classes of 2,000 notes in test set.

Correct ICPC chapter	Predicted ICPC chapter																		
	A	B	D	F	H	K	L	N	P	R	S	T	U	W	X	Y	Z	Sum	% correct
A	**13**	0	10	0	0	13	71	0	3	25	12	0	0	0	2	0	0	149	8.7 %
B	0	**0**	0	0	0	1	25	0	0	6	0	0	0	0	0	0	0	32	0.0 %
D	1	0	**64**	0	0	1	47	0	0	4	9	0	0	0	1	0	0	127	50.3 %
F	0	0	0	**19**	0	1	30	1	0	5	2	0	0	0	0	0	0	58	32.7 %
H	0	0	0	0	**16**	2	29	0	0	10	4	0	0	0	1	0	0	62	25.8 %
K	0	0	3	0	0	**158**	56	0	0	5	0	0	0	0	1	0	0	223	70.8 %
L	0	0	3	0	0	5	**348**	1	0	5	9	0	1	0	1	0	0	373	93.2 %
N	2	0	2	0	0	9	42	**4**	3	1	0	0	0	0	3	0	0	66	6.0 %
P	1	0	2	0	0	5	93	0	**33**	4	0	0	0	0	3	0	0	141	23.4 %
R	3	0	3	0	0	5	73	0	0	**170**	2	0	0	0	2	0	0	258	65.8 %
S	0	0	2	0	3	2	84	0	1	3	**128**	0	0	0	0	0	0	223	57.3 %
T	1	0	2	0	0	8	30	1	5	2	0	**2**	0	0	2	0	0	53	3.7 %
U	0	0	0	0	0	2	31	0	5	1	2	0	**1**	0	0	0	0	42	2.3 %
W	0	0	0	0	0	7	56	0	1	0	0	0	0	**15**	4	0	0	83	18.0 %
X	0	0	6	0	0	8	45	0	1	3	1	0	0	3	**23**	0	0	90	25.5 %
Y	0	0	1	0	0	2	14	0	1	0	0	0	1	0	0	**0**	0	19	0.0 %
Z	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	**0**	1	0.0 %

5 Discussion and Future Work

When considering the results, we must bear in mind that they are from a single run. To verify their validity they should be averaged over several test runs of independent samples.

Even though the accuracy varies a lot for the individual chapters, the results are still quite promising. The most notable feature is how the L (Muscoloskeletal) class appears to soak up the majority of the misclassified cases. We are not sure why this is happening. The L group constitutes the largest group in the training set, followed by the R, K and S groups. When attempting to perform the same classification task without the L cases the S group became the major misclassification bin, but in a less dramatic fashion; the overall accuracy rate rose to 57.5 %. In general, our naive, largely domain-ignorant approach granted results that are interesting enough to legitimate further work in this area.

There are several possible approaches to approving the predictive quality of the classifier. We made no attempts to normalize the vocabulary in the training data. Techniques such as stemming or mapping terms to a common controlled vocabulary would reduce the number of relevant features. This would also involve dealing with common misspellings [7] and dialect terms, both of which are quite common in our dataset. Wilcox [24] notes that the use of expert knowledge can provide a significant boost to medical text report classifiers. It would also be worth investigating if the use of accompanying information from the EPR, such as lab results and prescriptions, can help improve classification quality. Another possible approach is to view the encounter note in its longitudinal context by also considering notes from previous (and following) encounters.

We made no efforts to control the amount of noise in the classifiers or to screen the notes in the test data set. Very short notes and notes with non-standard language use were not discarded. Also, the influence of n-gram feature threshold selection on the quality of the results could have been evaluated. Similarly, the effect of using additional parameters such as average note length and n-gram partial coincidence would have been worth investigating.

The a priori anonymization could also influence the results. Since the anonymization tool only allows known non-sensitive words, it is likely that special and unusual words are lost. Such words may have a higher predictive effect than more common words. Comparing the classifier on a non-anonymized dataset could possibly indicate how much of destructive effect that is incurred due to anonymization.

The choice of ICPC chapter codes as class indicators is not necessarily a natural choice. Indeed, this may be seen as a simplification of the problem of diagnosis prediction. Alternatives include grouping according to ICPC component codes or, as a natural follow-up, attempting to classify into the full ICPC codeset of 726 different codes.

ACKNOWLEDGEMENTS

Thanks go to Amund Tveit, Ole Edsberg, Inger Dybdahl Sørby and Gisle Bjørndal Tveit for comments and suggestions.

REFERENCES

[1] Act of 18 may 2001 no. 24 on personal health data filing systems and the processing of personal health data, 2004.04.12 2001.

[2] D. B. Aronow, S. Soderland, J. M. Ponte, F. Feng, W. B. Croft, and W. G. Lehnert, 'Automated classification of encounter notes in a computer based medical record', *Medinfo*, **8 Pt 1**, 8–12, (1995).

[3] Elisabeth Bayegan, *Knowledge Representation for Relevance Ranking of Patient-Record Contents in Primary-Care Situations*, Ph.D. dissertation, Norwegian University of Science and Technology (NTNU), 2002.

[4] M. Fiszman, W. W. Chapman, D. Aronsky, R. S. Evans, and P. J. Haug, 'Automatic detection of acute bacterial pneumonia from chest x-ray reports', *J Am Med Inform Assoc*, **7**(6), 593–604, (2000). Evaluation Studies Journal Article.

[5] Pius Franz, Albrecht Zaiss, Stefan Schulz, Udo Hahn, and Rdiger Klar, 'Automated coding of diagnoses - three methods compared', in *Proceedings of the Annual Symposium of the American Society for Medical Informatics (AMIA)*, Los Angeles, CA, USA, (2000).

[6] Johannes Frnkranz, 'Round robin classification', *J. Mach. Learn. Res.*, **2**, 721–47, (2002).

[7] W. R. Hersh, E. M. Campbell, and S. E. Malveau, 'Assessing the feasibility of large-scale natural language processing in a corpus of ordinary medical records: a lexical analysis', *Proc AMIA Annu Fall Symp*, 580–4, (1997).

[8] W. R. Hersh, T. K. Leen, P. S. Rehfuss, and S. Malveau, 'Automatic prediction of trauma registry procedure codes from emergency room dictations', *Medinfo*, **9 Pt 1**, 665–9, (1998).
[9] I. M. Hofmans-Okkes and H. Lamberts, 'The international classification of primary care (icpc): new applications in research and computer-based patient records in family practice', *Fam Pract*, **13**(3), 294–302, (1996).
[10] B. Honigman, P. Light, R. M. Pulling, and D. W. Bates, 'A computerized method for identifying incidents associated with adverse drug events in outpatients', *Int J Med Inform*, **61**(1), 21–32, (2001). Journal Article.
[11] WONCA International, *ICPC-2: International Classification of Primary Care*, Oxford Medical Publications, 2 edn., 1998.
[12] N. L. Jain and C. Friedman, 'Identification of findings suspicious for breast cancer based on natural language processing of mammogram reports', *Proc AMIA Annu Fall Symp*, 829–33, (1997).
[13] Thorsten Joachims, 'Text categorization with suport vector machines: Learning with many relevant features', in *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pp. 137–142, London, UK, (1998). Springer-Verlag.
[14] Andries Kruger, C. Lee Giles, Frans Coetzee, Eric Glover, Gary Flake, Steve Lawrence, and Cristian Omlin, 'Deadliner: Building a new niche search engine', in *Ninth International Conference on Information and Knowledge Management, CIKM 2000*, Washington, DC, (2000).
[15] Leah S. Larkey and W. Bruce Croft, 'Combining classifiers in text categorization', in *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 289–97, Zurich, Switzerland, (1996). ACM Press.
[16] L. Letrilliart, C. Viboud, P. Y. Boelle, and A. Flahault, 'Automatic coding of reasons for hospital referral from general medicine free-text reports', *Proc AMIA Symp*, 487–91, (2000).
[17] Pavel Makagonov, Mikhail Alexandrov, and Alexander Gelbukh, 'Clustering abstracts instead of full texts', *Lecture Notes in Computer Science*, **3206**, 129–35, (2004).
[18] Alan D. March, Eitel J. M. Laura, and Jorge Lantos, 'Automated icd9-cm coding employing bayesian machine learning: a preliminary exploration', in *Simposio de Informtica y Salud 2004*, (2004).
[19] G. B. Melton and G. Hripcsak, 'Automated detection of adverse events using natural language processing of discharge summaries', *J Am Med Inform Assoc*, **12**(4), 448–57, (2005).
[20] G. Nilsson, H. Ahlfeldt, and L. E. Strender, 'Textual content, health problems and diagnostic codes in electronic patient records in general practice', *Scand J Prim Health Care*, **21**(1), 33–6, (2003). Journal Article.
[21] Y. Satomura and M. B. do Amaral, 'Automated diagnostic indexing by natural language processing', *Med Inform (Lond)*, **17**(3), 149–63, (1992).
[22] Amund Tveit, Ole Edsberg, Thomas Brox Røst, Arild Faxvaag, Øystein Nytrø, Torbjørn Nordgård, Martin Thorsen Ranang, and Anders Grimsmo, 'Anonymization of general practioner's patient records', in *Proceedings of the HelsIT'04 Conference*, Trondheim, Norway, (2004).
[23] Rodrigo F. Vale, Berthier A. Ribeiro-Neto, Luciano R.S. de Lima, Alberto H.F. Laender, and Hermes R.F. Junior, 'Improving text retrieval in medical collections through automatic categorization', *Lecture Notes in Computer Science*, **2857**, 197–210, (2003).
[24] A. B. Wilcox and G. Hripcsak, 'The role of domain knowledge in automating medical text report classification', *J Am Med Inform Assoc*, **10**(4), 330–8, (2003).

A Framework for the study of Evolved Term-Weighting Schemes in Information Retrieval

Ronan Cummins and **Colm O'Riordan** [1]

Abstract. Evolutionary algorithms and, in particular, Genetic Programming (GP) are increasingly being applied to the problem of evolving term-weighting schemes in Information Retrieval (IR). One fundamental problem with the solutions generated by these stochastic processes is that they are often difficult to analyse. A number of questions regarding these evolved term-weighting schemes remain unanswered. One interesting question is; do different runs of the GP process bring us to similar points in the solution space?

This paper deals with determining a number of measures of the distance between the ranked lists (phenotype) returned by different term-weighting schemes. Using these distance measures, we develop trees that show the phenotypic distance between these term-weighting schemes. This framework gives us a representation of where these evolved solutions lie in the solution space.

Finally, we evolve several global term-weighting schemes and show that this framework is indeed useful for determining the relative closeness of these schemes and for determining the expected performance on general test data.

1 INTRODUCTION

Information retrieval (IR) is concerned with the return of relevant documents from a collection of unstructured documents given a user need. It has been recognized that the effectiveness of vector space approaches to IR depend crucially on the term weighting applied to the terms of the document vectors [15]. These term-weights are typically calculated using term-weighting schemes that assign values to terms based on how useful they are likely to be in determining the relevance of a document. Documents are scored in relation to a query using one of these term-weighting schemes and are returned in a ranked list format.

Genetic Programming (GP) is a biologically inspired search algorithm useful for searching large complex spaces. Inspired by the theory of natural selection, the GP process creates a random population of solutions. These solutions, encoded as trees, undergo generations of selection, reproduction and mutation until suitable solutions are found. As GP is a non-deterministic algorithm it cannot be expected to produce a similar solution each time. Restart theory in GP suggests that it is necessary to restart the GP a number of times in order to achieve good solutions [9]. As a result, an important question regarding the solutions generated by the GP process is; do all the good solutions behave similarly or is the GP bringing us to a different area in the solution space each time?

Recently, IR fusion techniques, that use the rankings from several retrieval systems to determine the final document ranking, have been shown to increase the performance of IR systems [16]. These techniques only work when the ranked lists from the different retrieval systems return different ranked lists. Thus, when new term-weighting schemes are developed it is important, in many respects, to determine if these new schemes are similar to existing ones in terms of the ranked lists produced, or if indeed they belong to a new family of weighting scheme.

This paper presents a framework for evaluating the distance between the ranked lists produced from different term-weighting schemes in order to understand the relative closeness of these schemes. We develop two different distance measures and show that they are useful in determining how the term-weighting schemes are expected to perform in a general environment. We use these distance measures to create trees visualizing the distances between the weighting schemes.

Section 2 of this paper introduces term-weighting schemes useful for determining the discrimination value of a term. Section 3 introduces the GP process and existing approaches using GP to evolve term-weighting schemes are also discussed. Section 4 introduces our framework and outlines two distance measures. Our experimental setup is outlined in section 5 while section 6 discusses our results. Finally, our conclusions and future work are summarised in section 7.

2 INFORMATION RETRIEVAL

2.1 Term-Weighting for vector models

Term-weighting schemes assign values to terms based on measures of the term in both a global (collection-wide) and local (document-specific) context. Yu and Salton [19] suggest that the best distinguishing terms are those which occur with a high frequency in certain documents but whose overall frequency across a collection is low (low document frequency). They conclude from this that a term weighting function should vary directly with term frequency and inversely with document frequency. The *idf* scheme, first introduced by Sparck Jones [17], gives a higher weight to terms that occur in fewer documents. The original *idf* measure is often calculated as follows:

$$idf = log(\frac{N+1}{df_t}) \quad (1)$$

where N is the number of documents in the collection and df_t is the number of documents containing term t. A modern weighting scheme developed by Robertson et al. [13] is the BM25 weighting scheme. The global part of this weighting scheme is a variation of the traditional *idf* measure and is calculated as follows:

$$idf_{rsj} = log(\frac{N - df_t + 0.5}{df_t + 0.5}) \quad (2)$$

[1] University of Ireland, Galway. email: ronan.cummins@nuigalway.ie, colmor@it.nuigalway.ie

The idf measure forms the basis of many modern term-weighting schemes as it determines what initial weight a search term should receive [12]. It is worth noting that documents are typically not retrieved by idf only, and are usually used in conjunction with local measures to aid retrieval performance. However, if we can firstly find out what initial weight a search term should be given, we can then improve upon this by looking at the within-document characteristics to further improve retrieval performance. Developing global weighting schemes separately has been shown to benefit the performance of IR systems [11, 4, 14] and is an important goal in developing full weighting schemes which include local characteristics, like term-frequency and document normalisation. These idf type schemes are also used in many other domains within IR to weight features (e.g. document classification).

3 GENETIC PROGRAMMING

Genetic Programming [8] is a stochastic searching algorithm, inspired by natural selection. In the GP process, a population of solutions is created randomly (although some approaches seed the initial population with certain known solutions). The solutions are encoded as trees and can be thought of as the genotypes of the individuals. Each tree (genotype) contains nodes which are either functions (operators) or terminals (operands). Each solution is rated based on how it performs in its environment. This is achieved using a fitness function. Having assigned the fitness values, selection can occur. Individuals are selected for reproduction based on their fitness value. Fitter solutions will be selected more often.

Once selection has occurred, reproduction can start. Reproduction (recombination) can occur in variety of ways. Crossover is the main reproductive mechanism in GP. When two solutions are selected from the selection process, their genotypes are combined to create a new individual. One point crossover is the norm for genetic programming. This is where a single point is located in both parents and the sub-trees are swapped at these points to create two new solutions. Mutation (asexual reproduction) is the random change of the value of a gene (or the change of a subtree) to create a new individual.

Selection and recombination occurs until the population is replaced by newly created individuals. Once the recombination process is complete, each individual's fitness in the new generation is evaluated and the selection process starts again. The process usually ends after a predefined number of generations. Bloat is a common phenomenon in GP. Bloat is where solutions grow in size without a corresponding increase in fitness.

3.1 Phenotype

The phenotype of the individual is often described as its behaviour. Selection occurs based on the fitness only. Fitness is determined by the phenotype which is in turn determined by the genotype. As one can imagine, different genotypes can map to the same phenotype, and different phenotypes can have the same fitness. For most problems in GP in an unchanging environment, identical genotypes will map to identical phenotypes which will have the same fitness.

3.2 Previous Research

GP techniques have previously been adopted to evolve weighting functions and are shown to outperform standard weighting schemes in an adhoc framework [6, 10, 18, 4]. However, in many of these approaches a critical analysis of the solutions evolved is not presented. It is important to gain an understanding of the solutions obtained from these evolutionary processes and have a means of rating the differences between the schemes.

In [7], differences in retrieval systems are analysed using the ranked lists returned from the various systems. The distance between two ranked lists is measured using the number of out-of-order pairs. Using the measure it can then be determined if two systems are in essence the same (i.e. if they return the same ranked lists for a set of queries). Spearman's rank correlation and Kendall's tau are two common correlations that measure the difference between ranked sets of data. Both Spearman's rank correlation and Kendall's tau use all of the ranked data in a pair of ranked lists.

4 FRAMEWORK

4.1 Phenotypic Distance Measures

Figure 1 shows how the GP paradigm is adopted to evolve term-weighting schemes in IR. We use mean average precision (MAP) as our fitness function as it is a commonly used metric to evaluate the performance of IR systems and is known to be a stable measure [1]. Furthermore, it has been used with success in previous research evolving term-weighting schemes in IR [6, 18].

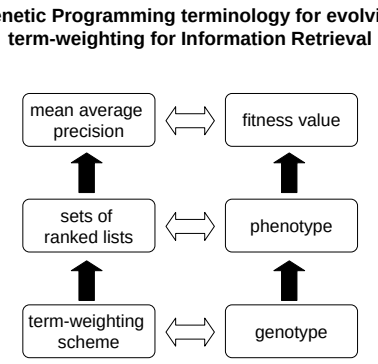


Figure 1. GP for Information Retrieval

For our framework, we measure the phenotype of our solutions by examining the sets of ranked lists returned by the term-weighting solution for a set of topics on a document collection (its environment). Spearman's rank correlation uses all available document ranks from two ranked lists and not just the ranks of relevant documents. We wish to develop distance measures for the parts of the ranked lists which affect the MAP (fitness) of a solution. This is important as the rank of relevant documents is the only direct contributing factor to the fitness of individuals within the GP.

To compare two sets of ranked lists, we introduce a measure which essentially measures the average difference between the ranks of relevant documents in two sets of ranked lists. In this measure, we ignore the ranks of non-relevant documents as they do not contribute to the fitness although they do technically contribute to the phenotype of the individual. This measure will tell us if the same relevant documents are being retrieved at, or close to, the same ranks and will tell us if the weighting schemes are evolving towards solutions

that promote similar features of relevant documents. Thus, one of the phenotypic distance measures ($dist(a,b)$), where a and b are two weighting schemes, is defined as follows:

$$\frac{1}{R}\sum_{i\in R}\begin{cases}|lim - r_i(b)| & \text{if } r_i(a) > lim\\ |r_i(a) - lim| & \text{if } r_i(b) > lim\\ |r_i(a) - r_i(b)| & \text{otherwise}\end{cases}$$

where R is the set of relevant documents in the collection for all of the queries used and $r_i(a)$ is the rank position of relevant document i under weighting scheme a. lim is the maximum rank position available from a list and is usually 1000 (as this is the usually the maximum rank for official TREC runs). As a result, relevant documents that are ranked outside the top 1000 are treated as being at rank 1000. Thus, when comparing two schemes this measure will tell us how many rank positions, on average, a relevant document is expected to change from scheme a to scheme b . Although different parts of the phenotype will impact on the fitness in different amounts (i.e. changes of rank for relevant documents at positions near 1000 do not significantly effect the MAP) they are an important part in distinguishing the behaviour of the phenotype. The change in position at high ranks can tell us about certain features of weighting scheme and the behaviour at these ranks.

We also develop a second measure of the distance between two ranked lists which takes into account the effect a change in rank has on MAP. To measure the actual difference a change in rank could make in terms of MAP, we modify the $dist(a,b)$ measure so that the change in rank of a relevant document is weighted on how it effects MAP. This weighted distance measure ($w\_dist(a,b)$) is similar to the measure described in [2] and is calculated as follows:

$$\frac{1}{Q}\sum_{q\in Q}\frac{1}{R_q}\sum_{i\in R_q}\begin{cases}\left|\frac{1}{lim} - \frac{1}{r_i(b)}\right| & \text{if } r_i(a) > lim\\ \left|\frac{1}{r_i(a)} - \frac{1}{lim}\right| & \text{if } r_i(b) > lim\\ \left|\frac{1}{r_i(a)} - \frac{1}{r_i(b)}\right| & \text{otherwise}\end{cases}$$

where Q is the number of queries and R_q is the relevant documents for a query q. This measure tells us how a change in rank of a relevant document will affect the MAP (i.e. changes of rank at positions close to 1000 will not change the MAP significantly, while changes of rank in the top 10 may change MAP considerably). Of course, it is entirely possible that two ranked lists could be considerably different yet have a similar MAP, as they may be promoting different relevant documents.

4.2 Neighbour-joining trees

Neighbour-joining is a bottom-up clustering method often used for the creation of phylogenetic trees. However, we use the method to produce trees that represent solutions that are from *different* runs of our GP. The algorithm requires knowledge of the distance between entities that are to be represented in the tree. A distance matrix is created for the set of entities using a distance measure and the tree can then be produced from the resulting data. We use this clustering technique to visualize the phenotypic distance between the best solutions output by our GP. For example, if we have N entities or solutions, we can create an $N \times N$ distance matrix using one of our distance measures. Then, using this distance matrix, we can then create a tree using a suitable drawing package [3] which represents the data and can provide a visualisation into where our solutions lie in relation to each other. This model is also well suited to our evolutionary paradigm. We use this technique simply to visualise the distance between our term-weighting solutions which are developed using GP.

5 EXPERIMENTAL SETUP

5.1 Approach Adopted

We evolve global term-weighting schemes in the following framework:

$$score(d,q) = \sum(gw_t \times qtf) \quad (3)$$

where $score(d,q)$ is the score a document d recieves in relation to a query q, gw_t is the global weighting and qtf is the frequency of the term in the query. All documents in the collection are scored in relation to the query and ranked accordingly. We are only evolving the global (term-discrimination) part of the weighting scheme as an example of our framework. However, the entirety of the term-weighting scheme can be evolved and analysed in a similar manner.

5.2 Training and Test Collections

We use collections from TREC disks 4 and 5 as our test collections. A different set of 50 TREC topics is used for each of the collections (apart from the Federal Register collection (FR) for which we use 100 TREC topics). For each set of topics we create a medium length query set (m), consisting of the title and description fields, and a long query set (l) consisting of the title, description and narrative fields. We also use documents from the OHSUMED collection as a test collection for medium length queries (OH90-91). We only use the topics in these sets that have relevant documents in the collection.

The TRAIN collection (used in training) consists of 35,412 documents from the OHSUMED collection and the 63 topics. The lengths of these topics range from 2 to 9 terms. Standard stop-words from the Brown Corpus[2] are removed and remaining words are stemmed using Porter's algorithm. No additional words are removed from the narrative fields as is the case in some approaches. Table 1 shows some characteristics of the document collections used in this research.

Table 1. Document Collections

Collection	#Docs	#words/doc	#Topics	medium	long
TRAIN	35,412	72.7	0-63	4.96	None
LATIMES	131,896	251.7	301-350	9.9	29.9
FBIS	130,471	249.9	351-400	7.9	21.9
FT91-93	138,668	221.8	401-450	6.5	18.7
FR	55,630	387.1	301-400	8.9	25.9
OH90-91	148,162	81.4	0-63	4.96	None

5.3 Terminal and Function Set

Tables 2 and 3 show the functions and terminals that are used in all runs of the GP.

5.4 GP parameters

We use MAP as our fitness function. All tests are run for 50 generations with an initial random population of 100 solutions on the training collection (TRAIN) detailed in Table 1. The tournament size is set to 3. We restrict all trees to a depth of 6. As a result this has the effect of reducing bloat, improving generalisation, reducing the search space and increasing the speed of the GP. As our highest order operator is binary, the longest individual we can have can contains 63

[2] http://www.lextek.com/manuals/onix/stopwords1.html

Table 2. Function Set

Function	Description
+, ×, /, -	arithmetic functions
log	the natural log
$\sqrt{}$	square-root function
sq	square

Table 3. Terminal Set

Terminal	Description
N	no. of documents in the collection
df	document frequency of a term
cf	collection frequency of a term
V	vocabulary of collection (no. of unique terms)
C	size of collection (total number of terms)
0.5	*the constant* 0.5
1	*the constant* 1
10	*the constant* 10

genes $(2^6 - 1)$. We believe that this is a large enough space in which to find suitable term-weighting schemes. The creation type used is the standard ramped half and half creation method used by Koza [8]. We use an elitist strategy where the best individual is automatically transfered to the next generation. 4% mutation is used in our experiments. Due to the stochastic nature of GP a number of runs is often needed to allow the GP converge to a suitably good solution. We run the GP seven times and choose the best solution from each of those runs. This gives us seven evolved solutions and two benchmark solutions (1) (2) to use with our document collections.

6 RESULTS

Table 4 shows the MAP of seven global evolved weighting schemes (gw) on our training data in no particular order. We can see that all the evolved schemes are better than our benchmarks (idf and idf_{rsj}) in terms of MAP.

Table 4. % MAP for all global weightings

Collection	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
TRAIN	19.83	19.98	22.05	21.98	21.60	21.69	20.11	20.11	20.75

Figure 2 shows the best and average of the population from the two best runs of the GP (i.e. gw_1 and gw_2). It is worth noting that the best individual from the seven randomly created populations (i.e. generation 0) is not better than the best solution produced after the 50^{th} generation from the worst of the seven runs.

Tables 5, 6 and 7 shows the distance matrices for all the global weighting schemes for the training data using Spearman's rank correlation, $dist(a, b)$ and $w\_dist(a, b)$ measures respectively. Spearman's rank correlation gives us values in the range of -1 to $+1$ and uses all of the documents in the ranked list. As the Spearman correlations of the ranked lists produced by the global weighting scheme are all positively correlated, we simply use $1-$ Spearman's rank correlation as a distance measure. This will give us 1 if the lists are randomly correlated and 0 if they are identical. We use this correlation

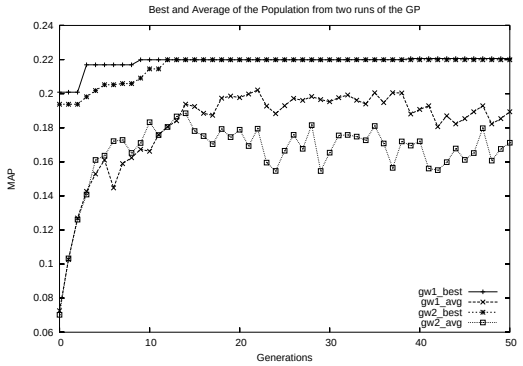


Figure 2. Best and Average Fitness for best two global runs

as a comparison to our distance measures that only look at distances of relevant documents.

The values in Table 6 indicate the average number of rank positions a relevant document changes. While the values in Table 7 indicate the maximum possible percentage MAP difference between two schemes. By looking at the difference between the ranked lists of each global weighting we can get an idea of the landscape of the solution space in the global domain.

Table 5. $1-$ spearman's rank correlation between all global weightings on TRAIN

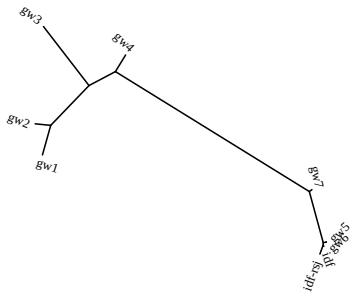
Scheme	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
idf	00.00	0.014	0.524	0.501	0.503	0.396	0.007	0.007	0.078
idf_{rsj}		00.00	0.531	0.507	0.507	0.400	0.020	0.020	0.089
gw_1			00.00	0.059	0.186	0.174	0.523	0.523	0.451
gw_2				00.00	0.210	0.136	0.499	0.499	0.425
gw_3					00.00	0.170	0.501	0.501	0.435
gw_4						00.00	0.393	0.393	0.324
gw_5							00.00	00.00	0.076
gw_6								00.00	0.076
gw_7									00.00

Table 6. $dist$ measure between all global weightings on TRAIN

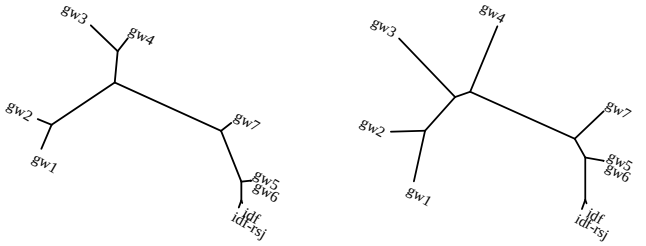
Scheme	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
idf	00.00	01.27	36.50	35.07	32.81	30.31	03.65	03.65	11.15
idf_{rsj}		00.00	35.94	34.32	31.90	29.42	04.52	04.52	12.01
gw_1			00.00	05.07	21.26	18.46	35.34	35.34	28.13
gw_2				00.00	20.60	15.96	34.15	34.15	27.08
gw_3					00.00	06.62	30.48	30.48	24.66
gw_4						00.00	28.37	28.37	22.15
gw_5							00.00	00.00	09.16
gw_6								00.00	09.16
gw_7									00.00

Table 7. w_dist % measure between all global weightings on TRAIN

Scheme	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
idf	00.00	00.21	04.30	04.17	04.10	03.91	00.99	00.99	01.78
idf_{rsj}		00.00	04.20	04.15	04.26	04.01	01.10	01.10	01.86
gw_1			00.00	01.33	02.73	03.16	04.23	04.23	04.25
gw_2				00.00	02.50	02.16	04.09	04.09	04.03
gw_3					00.00	02.64	03.96	03.96	03.95
gw_4						00.00	03.48	03.48	03.38
gw_5							00.00	00.00	01.18
gw_6								00.00	01.18
gw_7									00.00



1−Spearman's rank correlation



dist w_dist

Figure 3. Neighbour-Joining trees for global weightings

Firstly, from Figure 3 we can see that the phenotypic distance measures produce trees of a similar structure. The only difference in form is that gw_3 and gw_4 are clustered together directly using the unweighted $dist$ measure. It is important to note that the trees visualize different aspects of the ranked lists. For example, the distance between the top four performing schemes (gw_1 to gw_4) and the remaining schemes is greater in the tree created from Spearman's rank correlation than for the other two trees. This is because Spearman's rank correlation uses the ranks of non-relavant documents. Looking at the tree produced by the $dist(a, b)$ measure, we can see that gw_3 and gw_4 are quite similar in terms of the actual ranks of relevant documents. However, when looking at the tree produced by $w\_dist(a, b)$ for these two schemes, we can see that some of these differences are at low ranks as the possible difference in MAP is quite large.

In general, we can see that idf_{rsj}, idf, gw_5 and gw_6 are phenotypically close. Schemes gw_5 and gw_6 are actually phenotypically equivalent (i.e. return the same ranked lists) but not genotypically equivalent. The two versions of idf are very close. Schemes gw_1 and gw_2 are also phenotypically close while gw_3 and gw_4 are somewhat similar. An important point to note is that as we get phenotypically further from the best solution (gw_1) we see a relative drop in MAP on our training collection. This indicates that the solutions are evolving towards the ranked lists (on the training set) that are produced by gw_1. Obviously, phenotypically close solutions will have a similar fitness but it is not neccessarily true that solutions with a similar fitness will have a similar phenotype (e.g. as one can imagine that there exists many poor performing functions which return equally bad but different ranked lists). It is worth noting that these trees should be produced from the training data as this is the environment where the solutions were evolved. However, these trees can help us to predict the behaviour of the schemes on general data (if our training data is a representative sample).

Tables 8 and 9 show the MAP of all schemes for unseen test data on medium and long queries. Firstly, we can see that the differences in MAP between the evolved weightings and idf_{rsj} are all statistically significant ($p < 0.05$) using a two-tailed t-test. Both version of idf perform similarly as expected. We can see that gw_1 is no longer the best evolved weighting scheme, although it is still significantly better than idf. Schemes gw_2, gw_3 and gw_4 are now the best performing schemes on most of the collections. Schemes gw_5 and gw_6 still perform only slightly better than idf_{rsj}, while gw_7 still performs slightly better than these again. It would seem that gw_1 has overtrained slightly on the training collection. It is also worth pointing out that our training set seems to be quite general as most of the schemes perform similarly on test data. If we look at the genotypes of some of the schemes it leads us to a similar conclusion. We have re-written the following formulas in a more intuitive manner to provide transparency to the process. As a result, the re-written formulas may also be shorter (in depth) that those that were evolved originally.

$$gw_1 = \frac{V^2 cf^2 \sqrt{cf}}{C.df^3} + \sqrt{cf} \qquad gw_2 = \frac{cf^2 \sqrt{cf}}{df^3}$$

$$gw_3 = \sqrt{(log(\frac{cf}{df}))^2 \times \frac{N}{df} \times (\frac{N^2}{df} + 1)}$$

$$gw_4 = \sqrt{\frac{cf^3 N}{df^4}} \qquad gw_5 = \sqrt{\sqrt{\frac{0.5}{df}}}$$

$$gw_6 = \sqrt{\frac{\sqrt{df}}{df}} \qquad gw_7 = \sqrt{\frac{\sqrt{cf/N}}{df^2}}$$

We can see that gw_1 is a more specific form of gw_2. Schemes gw_5 and gw_6 are an example of two different genotypes producing the same ranked lists. gw_6 will produce a score that is always double that of gw_5. We are evolving towards a ranked list on the training collection that is produced by the best two schemes (gw_1 and gw_2). The gw_2 scheme is a more general form of gw_1 and performs consistently better on our test data. The gw_3 scheme contains a problematic $log(cf/df)$ that will assign certain low frequency terms a zero weight [5] and makes it a poor choice for weighting in a retrieval context. This can be seen on the results for the FR collection when compared to one of its nearest neighbours gw_4. When looking at the individual queries for this collection (FR), we have determined that the difference between gw_4 and the other top schemes (gw_1 to gw_3) is only large for a very small number of queries. As a result it can be

Table 8. % MAP for idf and global weightings for Medium Queries

Collection	Topics	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
LATIMES	301-350 (m)	19.11	19.16	21.80	22.49	23.48	22.98	20.92	20.92	21.12
FBIS	351-400 (m)	10.30	10.41	15.16	15.68	14.55	14.33	11.61	11.61	11.72
FT91-93	401-450 (m)	27.38	28.15	27.52	27.86	27.56	27.92	27.04	27.04	27.10
FR	301-400 (m)	25.87	24.89	25.12	25.71	21.31	28.72	25.49	25.49	27.39
OH90-91	0-63 (m)	21.68	21.72	24.96	25.69	25.02	25.28	22.96	22.96	23.68
$\approx p\text{-}value$	241 Topics	0.272	-	0.004	0.0001	0.0001	0.0001	0.018	0.018	0.021

Table 9. % MAP for idf and global weightings for Long Queries

Collection	Topics	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
LATIMES	301-350 (l)	13.57	13.79	21.60	24.27	24.78	24.30	16.37	16.37	16.63
FBIS	351-400 (l)	06.76	06.97	12.30	13.32	14.07	13.84	08.34	08.34	09.01
FT91-93	401-450 (l)	23.11	23.13	27.17	28.28	28.31	29.13	24.95	24.95	25.80
FR	301-400 (l)	16.23	16.95	22.78	22.75	20.86	27.83	19.84	19.84	19.92
$\approx p\text{-}value$	241 Topics	0.300	-	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

concluded that gw_4 promotes certain useful features that are different than those of the rest of the schemes. These differences are noticeable on the FR collection because of its makeup. The gw_4 scheme seems to be a particularly robust global weighting scheme as shown on the test data. The difference between gw_4 and gw_3, for example, is not statistically significant. However, we know that gw_4 has advantagous retrieval features (as seen on the FR collection) for certain (albeit few) queries.

7 CONCLUSION

We have introduced two metrics that measure the distance between the ranked lists returned by different term-weighting schemes. These measures are useful for determining the closeness of term-weighting schemes and for analysing the solutions without the need to analyse the exact form (genotype) of a term-weighting scheme. This framework can be used for all types of term-weighting schemes and also fits well into the genetic programming paradigm.

The distance matrices produced from these distance measures can be used to produce trees that aid visualization of the solution space. The trees produced are also useful in determining the relative performance of the solutions on general test data. We have also shown that all the evolved global weighting schemes produced are evolving to a area of the solution space that is different from the types of idf currently being used to measure the discrimination value of a term. In future work, we intend to apply this framework to analyse entire term-weighting schemes which have been evolved.

ACKNOWLEDGEMENTS

This work is being carried out with the support of IRCSET (the Irish Research Council for Science, Engineering and Technology) under the Embark Initiative.

REFERENCES

[1] Chris Buckley and Ellen M. Voorhees, 'Evaluating evaluation measure stability', in *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 33–40, New York, NY, USA, (2000). ACM Press.

[2] Ben Carterette and James Allan, 'Incremental test collections', in *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 680–687, New York, NY, USA, (2005). ACM Press.

[3] Jeong-Hyeon Choi, Ho-Youl Jung, Hye-Sun Kim, and Hwan-Gue Cho, 'Phylodraw: a phylogenetic tree drawing system.', *Bioinformatics*, **16**(11), 1056–1058, (2000).

[4] Ronan Cummins and Colm O'Riordan, 'An evaluation of evolved term-weighting schemes in information retrieval.', in *CIKM*, pp. 305–306, (2005).

[5] Ronan Cummins and Colm O'Riordan, 'Evolving general term-weighting schemes for information retrieval: Tests on larger collections.', *Artif. Intell. Rev.*, **24**(3-4), 277–299, (2005).

[6] Weiguo Fan, Michael D. Gordon, and Praveen Pathak, 'A generic ranking function discovery framework by genetic programming for information retrieval', *Information Processing & Management*, (2004).

[7] P. Kantor, K. Ng, and D. Hull. Comparison of system using pairs-out-of-order, 1998.

[8] John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.

[9] Sean Luke, 'When short runs beat long runs', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 74–80, San Francisco, California, USA, (7-11 2001). Morgan Kaufmann.

[10] N. Oren, 'Re-examining tf.idf based information retrieval with genetic programming', *Proceedings of SAICSIT*, (2002).

[11] A. Pirkola and K. Jarvelin, 'Employing the resolution power of search keys', *J. Am. Soc. Inf. Sci. Technol.*, **52**(7), 575–583, (2001).

[12] S. E. Robertson and S. Walker, 'On relevance weights with little relevance information', in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 16–24. ACM Press, (1997).

[13] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau, 'Okapi at TREC-3', in *In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST*, (1995).

[14] Dmitri Roussinov, Weiguo Fan, and Fernando A. Das Neves, 'Discretization based learning approach to information retrieval.', in *CIKM*, pp. 321–322, (2005).

[15] Gerard Salton and Chris Buckley, 'Term-weighting approaches in automatic text retrieval', *Information Processing & Management*, **24**(5), 513–523, (1988).

[16] Alan F. Smeaton, 'Independence of contributing retrieval strategies in data fusion for effective information retrieval.', in *BCS-IRSG Annual Colloquium on IR Research*, (1998).

[17] Karen Sparck Jones, 'A statistical interpretation of term specificity and its application in retrieval', *Journal of Documentation*, **28**, 11–21, (1972).

[18] Andrew Trotman, 'Learning to rank', *Information Retrieval*, **8**, 359 – 381, (2005).

[19] C. T. Yu and G. Salton, 'Precision weighting - an effective automatic indexing method', *Journal of the ACM*, **23**(1), 76–88, (1976).

LexiRes: A Tool for Exploring and Restructuring EuroWordNet for Information Retrieval

Ernesto William De Luca and **Andreas Nürnberger** [1]

Abstract. The problem of word sense disambiguation in lexical resources is one of the most important tasks in order to recognize and disambiguate the most significant word senses of a term. Lexicographers have to decide how to structure information in order to describe the world in an objective way. However, the introduced distinctions between word meanings are very often too fine grained for specific applications. If we want to use or even combine lexical resources within information retrieval systems, for example, we might want to apply the lexical resources in order to disambiguate documents (retrieved from the web within an information retrieval system) given the different meanings (retrieved from lexical resources) of a search term having unambiguous description. Therefore, we are usually interested in a small list of meanings with very distinctive features. Since many lexical resources, especially WordNet, provide frequently too fine grained word sense distinctions, we implemented the tool LexiRes that gives the possibility to navigate lexical information, helping authors of already available lexical resources in deleting or restructuring concepts using automatic merging methods.

1 Introduction

Standard keyword based search engines retrieve documents without considering the importance of user oriented information presentation. It means that the user has to analyze every document and decide himself which are the documents that are relevant with respect to the context of his search. For example, users have to navigate every document in order to recognize to which meaning of their query words the documents belong to. Thus, it would strongly support a user if the context - which is defining the meaning of a word - could be recognized automatically and the documents could be labelled or grouped with respect to the meaning of the respective search terms. One way to obtain a context description of different word senses is to explore lexical resources using the word we are looking for in order to select concepts based on the linguistic relations of the lexical resource that define the different word senses. Such disambiguating relations are intuitively used by humans. However, if we want to automate this process, we have to use resources - such as probabilistic language models or ontologies - that define appropriate relations. One of these most important resources available to researchers for this purpose is WordNet [4] and its variations like MultiWordNet [3] and EuroWordNet [15] as discussed in the following.

However, since many lexical resources or ontologies, especially WordNet, provide frequently too fine grained word sense distinctions, we implemented the tool LexiRes that gives the possibility to navigate lexical information, helping authors of already available lexical resources in deleting or restructuring concepts using automatic merging methods. The restructured information can be navigated and explored. Authors can decide if word senses are unambiguous and important enough to let them in the hierarchy at the same place or if they express similar concepts and can be merged under the same (now, more general) meaning.

In the following, we first briefly introduce the structure of WordNet and EuroWordNet. Then we discuss the problem of word sense disambiguation in information retrieval and problems related to WordNet in order to motivate the LexiRes system, which is then presented in Sect. 4.

2 WordNet

WordNet [4] was designed by use of psycholinguistic and computational theories of human lexical memory. It provides a list of word senses for each word, organized into synonym sets (SynSets), each representing one constitutional lexicalized concept. Every element of a SynSet is uniquely identified by an identifier (SynSetID). It is unambiguous and carrier of exactly one meaning. Furthermore, different relations link these elements of synonym sets to semantically related terms (e.g. hypernyms, hyponyms, etc.). All related terms are also represented as SynSet entries. These SynSets also contains descriptions of nouns, verbs, adjectives, and adverbs. With this information we can describe the word context. Fig. 1 represents an example of the ontology hierarchy defined by WordNet [4]. This resource can be used for text analysis, computational linguistics and many related areas.

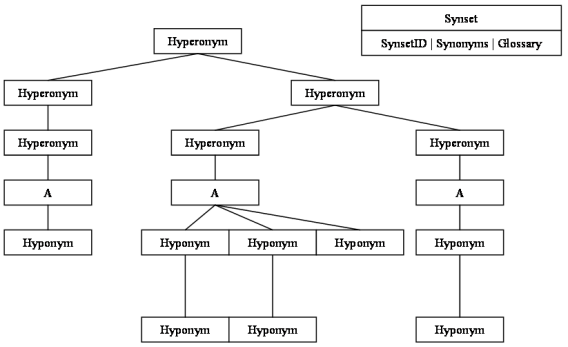


Figure 1. Example of an ontology hierarchy for a given term A.

[1] University of Magdeburg, Germany, email: deluca@iws.cs.uni-magdeburg.de

2.1 EuroWordNet

WordNet was first developed only for the English language. Then different versions were developed for several other languages as for example EuroWordNet [15] for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian). Given that we want to retrieve from the web different documents in different languages analysing different contexts, we decided to use the EuroWordNet multilingual lexical database. Its structure is the same as the Princeton WordNet [4] in terms of SynSets with different semantic relations between them. Each individual wordnet represents a unique language-internal system of lexicalizations. The Inter-Lingual-Index (ILI) was introduced in order to connect the WordNets of the different languages. Thus, it is possible to access the concepts (SynSets) of a word sense in different languages.

In addition to the Inter-Lingual-index, there is also a Domain-Ontology and a Top-Concept-Ontology related to this lexical database. The shared Top-Ontology is a superordinate hierarchy of 63 semantic distinctions for the most important language independent concepts (e.g. Artifact, Natural, Cause, Building) and is interconnected with the ILI through the WordNet-Offsets. Hereby a common semantic framework for all the languages is given, while language specific properties are maintained individually. The Domain-Ontology was created for use in information retrieval settings in order to obtain specific concepts (only implemented exemplary for the computer terminology). Figure 2 gives an overview over the architecture of the EuroWordNet whereby the single components and its relations are represented among one another.

3 Word Sense Disambiguation in Information Retrieval

User studies have shown that categorized information can improve the retrieval performance for a user. Thus, interfaces providing category information are more effective than pure list interfaces for presenting and browsing information [2]. The authors of [2] evaluated the effectiveness of different interfaces for organizing search results. Users strongly preferred interfaces that provide categorized information and were 50% faster in finding information organized into categories. Similar results based on categories used by Yahoo were presented in [7].

The tool which we present in this paper, was developed as part of our work research towards a (multilingual) retrieval system that classifies documents with respect to the search terms in unambiguous classes, so-called Sense Folders. The main idea of our approach is to provide additional disambiguating information to the documents of a result set retrieved from a search engine in order to enable to restructure or filter the retrieved document result set. The use of web documents implies an on line categorization approach of the documents given the query terms provided from the user. Thus, we can support the user in choosing the relevant information by categorizing the documents using different classification techniques. In the system presented in [8, 10], we use user and query specific information in order to annotate - and thus categorize - search results from other search engines or text archives connected to the meta search engine by web services. The system currently supports methods to group documents based on semantic disambiguation of query terms using an ontology that can be selected by the user. The system analyzes every search term and extracts the belonging SynSets, that are, the sets defining the different meanings of a term and the linguistic relations from the used ontology. Based on these terms, prototypical word vectors of the disambiguating classes (”Sense Folders” [8]) are constructed. Every document is assigned to its nearest prototype (computed by using the cosine similarity) and afterward this classification is revised by a clustering process.

Agreeing with [16] we see one document having one sense per collocation and discourse. But differentiating us from [16], we do not want to learn and disambiguate word senses from untagged corpora.

The idea of this approach is to use ontologies in order to disambiguate query terms used in the retrieved documents [9]. Thus it is possible to categorize documents with respect to the meaning of a search term, i.e. each document is assigned to the best matching meaning (”Sense Folder”) of the search terms used in it. Obviously, only one sense per document can be distinguished in this setting, which is, however, appropriate for many typical retrieval problems where only short documents are considered as, for example, in Web Search.

For this annotation process we currently use WordNet (resp. EuroWordNet). However, if we analyze it, different problems have to be resolved. Very often meanings are distinguished that are semantically very close. For example, searching for the term ”bank” in an information retrieval environment, the user usually wants to know if the retrieved documents belong to the meaning ”bank” in the sense of ”furniture” or in the sense of ”banking”. The fine grained linguistic differentiation between the ”depository bank” meaning and the ”building bank” one is very often not so significant in order to select a relevant document.

This problem of too fine grained description of meanings in WordNet makes on the one hand the automatic categorization very difficult and on the other hand burdens the users with a much too detailed specialization. Therefore, we propose a simple pruning strategy in order to obtain a reduced set of (more expressive) concepts for the categorization approach (see Sect. 3.2). Furthermore, we describe in the following some further problems that should be tackled for a better expressiveness of WordNet.

3.1 Problems of the EuroWordNet Hierarchy

In the following we briefly examine the main semantic limitations of WordNet and describe some problems that have to be solved for its better expressiveness (see also [6, 5, 13]).

Some lexical links of WordNet should be interpreted using formal semantics in order to express ”things in the world”. The authors of [13] revise the Top Level of WordNet (upper or general level) where the criteria of identity and unity are very general, in order to recognize the constraint violations occurring in it. The concepts of identity and unity are described in [13].

However, we analyze the expressiveness of every SynSet in order to better categorize the context for clustering purposes. It means that we merge categories that are in the same domain and that are not much different from another. This decision is based on our need of few unique classes that are carrier of an expressive meaning for a user as well as for an improved clustering performance.

An example is given in [10]. If we retrieve a word from WordNet, several meanings are assigned to the domain ”Factotum” that could be described as the class ”other domain, generic”. The reason for this assignment is simply the problem that the WordNet authors have to assign a domain to each SynSet. If a term can not be categorized (by the author) to a more specific domain, the generic domain ”Factotum” is used. Therefore, if we want to categorize documents with WordNet senses, we have to choose which senses are relevant and which are not, in order to obtain appropriate disambiguation results.

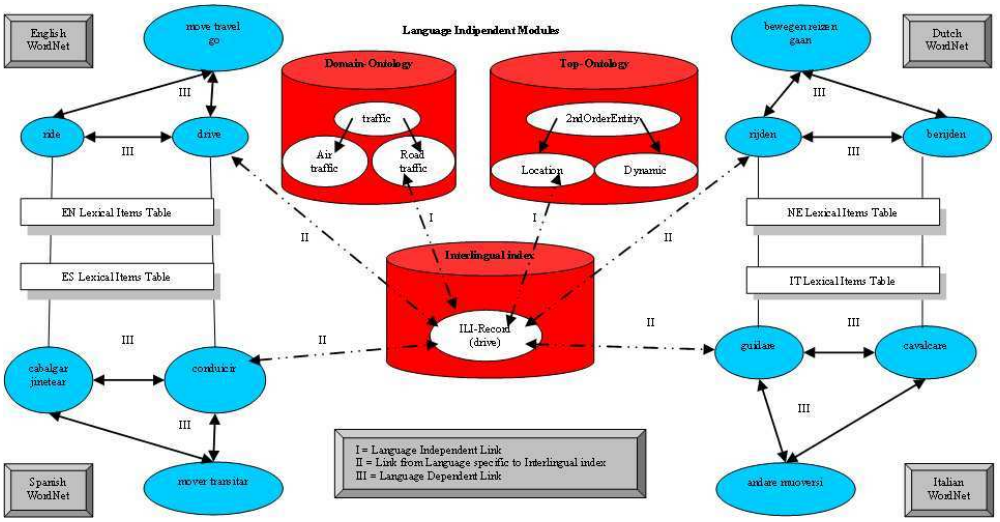


Figure 2. EuroWordNet Architecture (see [15]).

However, if we maintain all senses that are labelled with ”Factotum”, we have in many cases to distinguish between only slightly different contexts defined by different SynSets. One possibility to derive terms that have a very similar meaning is to analyze their hyponyms or hypernyms. If there are two senses described in WordNet belonging to the same domain, they often have the same hyponyms or hypernym. This frequently causes disambiguation problems that can not be solved if we keep all classes. For this reason, we decided to exclude some irrelevant (for the context disambiguation process) ”Factotum” SynSets.

Another critical point is given by the confusion between concepts and instances resulting in an ”expressivity lack” [5]. For example, if we look for the hyponyms of ”mountain” in WordNet, we will find the ”Olympus mount” as a subsumed concept of the word treated as ”volcano” and not as instance of it. Thus, we do not have a clear differentiation between what we use to describe (concepts) and their instantiation (instances). We also have the problem that we can not use only concepts or only instances because there is no intended separation between them in WordNet.

The authors of [12] treat also the important difference between endurance and perdurance of the entities that should be included in WordNet. Enduring and perduring entities are related to their behaviour in time. Endurants are always wholly present at any time they are present. Perdurants are only partially present, in the sense that some of their proper parts (e.g., their previous phases) may be not present. However, these aspects of instances are not discussed in this paper since they seem to be of less importance for the considered disambiguation problem.

When we deal with EuroWordNet, these problems persist, and other problems come along. The problem of automatically finding multilingual translation of word senses over languages can be solved using such a resource. The use of the Inter-Lingual-Index helps for this purpose, but the coverage of language-dependent word senses varies from language to language. The number of Synsets varies from an amount of 20.000 (german) to 150.000 (english) Synsets. Using this lexical resource, we have to take into account the missing (or incomplete) translations contained in the lexical resource, apart from the lexical gaps (word senses that exist in a language and not in another).

3.2 Merging the EuroWordNet SynSets

One possible way to tackle some of the problems described above is to merge SynSets manually, when the author means that they belong together. Another possibility is to use methods that restructure EuroWordNet by merging SynSets that have a very similar meaning. Therefore, we studied methods in order to automatically merge SynSets based on the analysis of the linguistic relations defined in EuroWordNet.

We implemented four online methods to merge SynSets based on relations like hypernyms and hyponyms, and further context information like glosses and domain. The first merging approach is based on context information extracted from the hypernymy relation (superordinate words) in order to define the Sense Folders. It means that we first build word vectors for every word sense (Sense Folder), containing the whole hypernymy hierarchy related to the query word. Then we compare all Sense Folders with one another and merge them when the similarity exceeds a given threshold (i.e., when their word vectors are sufficiently close to each other). A similar approach is applied for the hyponyms (subordinate words). In the third approach we merge the Sense Folders if their linguistic relations and context information (glosses) are similar. The fourth approach exploits the domain concept of MultiWordNet [3]. Here we merge the Sense Folders only if they belong to the same domain (having exactly the same domain description).

An evaluation of this methods was done on a small corpus of 252 documents retrieved from web searches that had been manually annotated. Hereby, we compared the manual annotated classes with the Sense Folders assigned using the approach described in [8] together with the merging functions implemented. Based on this first evaluation, the hypernym approach seemed to nicely merge Sense Folders that had similar hypernyms which even might be labeled with different domain descriptions. However, a better classification was

obtained for words that had fewer meanings (SynSets) before merging starts. The second approach based on hyponyms almost never merged SynSets due to the usually very different hyponyms assigned to each sense. Using the third approach, a lattice was built between the merged Sense Folders. This approach merges SynSets not having the same hypernyms, but similar words given from the descriptions of all relations and words together. With the fourth approach we are sure to merge Sense Folders that belong to the same context, describing it in a different way. The classification was always the best, but the Factotum problem as discussed in Sect. 3.1 persisted. If this merged class contains very different meanings and is used for classification, this classification is worse than before. The possibility to exclude such classes (labeled with the "Factotum" domain) will be studied in future work, e.g. by analyzing approaches that exploits combined information from the first three merging methods. For details of the evaluation see [11].

4 The lexical restructuring tool (LexiRes)

The main idea of this tool is to give authors the possibility to navigate the ontology hierarchy in order to restructure it, by manual merging or using the merging functions described in Section 3.2.

4.1 Related Work

Different work has been already done using the variants of WordNet. The authors of [1] developed VisDic for browsing and editing multilingual information taken from EuroWordNet. Here users can browse static information on text blocks.

Another web interface for multilingual information browsing is presented in [14]. Here a parallel corpus annotated with MultiWordNet [3] can be browsed as well as the words with their related annotated word senses, but the corpus is very restricted. All accessible information is static. This interface is used only for a bilingual search in a closed domain.

Other work dealing with the lexicography has shown that researchers in this area mostly deal with multilingual lexical resources or corpora only, without the possibility of merging similar word senses.

Given that the EuroWordNet format is defined by the EuroWordNet Database Editor Polaris that uses a proprietary specification, we first converted the EuroWordNet Database in an XML format, in order to access it with standard XML query tools. In order to retrieve information from this resource, we use the Exist Open Source native XML database.

4.2 The tool

In order to use the LexiRes tool, we have to load an ontology into its scratch framework. The tool currently supports the EuroWordNet structure, but can easily be extended for different ontologies. Considering that we use a multilingual lexical resource, we give the possibility to define the language we want to work with and the linguistic relations we want to show for recognizing the query word in the context menu. After having set it the hierarchy will be displayed.

Figure 3 shows a screenshot of the LexiRes editor. On the left side, we can enter the query words. On the right side, we can choose which collection we want to retrieve and which language we want to use as a source language. Looking for "bank", in the english language, the ontology engine retrieves 19 meanings. These meanings are describing the different word senses. Every word sense is represented as a SynSet. We can apply different actions for these SynSets. Some meanings that belong to the same domain, as the two "bank" - SynSets under the superordinate "incline" SynSet could be merged. If authors decide that the description of these SynSets is too fine grained, they can choose to merge the "source" SynSets to a "target". The goal is to obtain only word senses describing contexts as unambiguous as possible. Based on the merging a new SynSet is created to which all relations of the original SynSets are assigned. Authors can also decide that a SynSet should not be a carrier of meaning for the intended application of the ontology; this SynSet can be removed just clicking on it and choosing to remove it.

The linguistic relations as also the properties of every SynSet can be shown just picking the corresponding fields. These can be first set within the check boxes under the "show relations" area. If the author activates the check boxes, the linguistic relations related to the selected SynSet will be shown. The author can choose to "show properties" or "hide properties" with a right mouse click on a SynSet. Here all SynSet-related information is shown. The original XML code part of the SynSet can also be chosen clicking on the right mouse button and choosing the "show XML" option. The properties and the XML code are shown on the right side down of the interface under "Details".

The SynSets can be also automatically retrieved and translated in the different languages available in the ontology (see Figure 4). These can be set within the menu button language and can be shown, always SynSet-dependent within a click. We can notice that not all SynSet have a translation, due to the missing entries in the lexical resource.

As we said before, the tool gives the possibility to manually merge SynSets, when the authors decide that two SynSets belong to the same meaning and/or describe the same concept. The author working with LexiRes can also use an automatically created list of candidate SynSets that can be merged. This list can be created with the approaches discussed in 3.2. The system proposes the list of changes and the user can select to accept all or check each proposal for merging manually. At the moment these merging methods are implemented outside the tool. The resulting list of possible merging SynSets is first examinated from the authors and then done manually. After having restructured the ontology hierarchy, a new set of SynSets is created. This set is supposed to contain only word senses that are carrier of a distinctive meaning in the context of the considered application. This is a very important step for a use of lexical resources in information retrieval. The possibility to merge SynSets in advance gives the advantage to categorize the retrieved documents disambiguating them with structured word senses that facilitate an automatic classification process [8]. A detailed description of the evaluation of the automatic merging methods applied to the WordNet SynSets in given in [11].

5 Conclusions

In this paper we motivated and presented LexiRes, a tool to help lexicographers in exploring available lexical resources for navigating and restructuring them, especially for use in information retrieval frameworks. Furthermore, we have discussed how lexical resources, here EuroWordNet, can be used in order to disambiguate documents (retrieved from the web within an information retrieval system) given different meanings (retrieved from lexical resources). After having discussed the problems related to the EuroWordNet structure, we presented the functionality of our tool. Using LexiRes we obtain a hier-

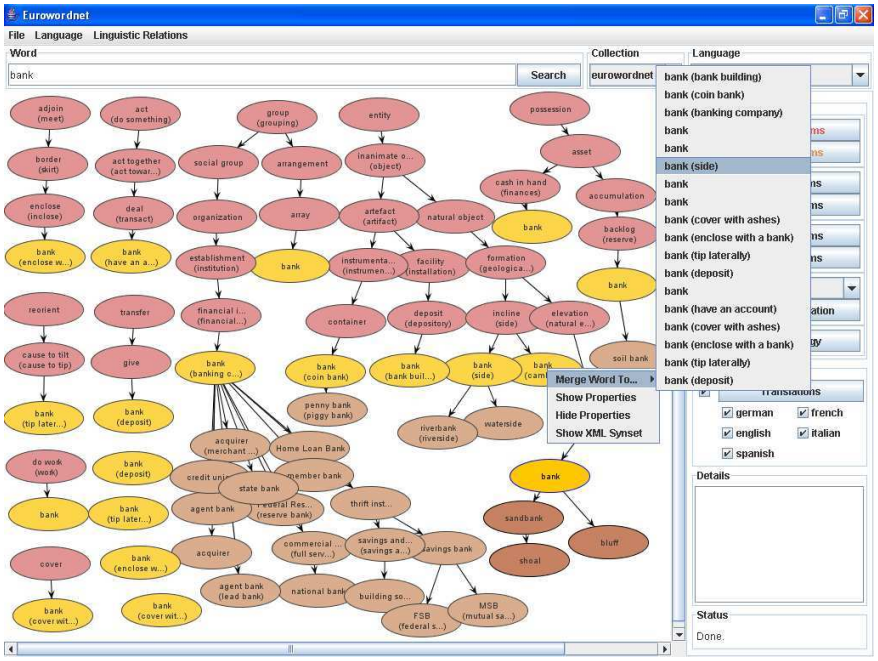


Figure 3. Example of the word ”bank” - manual merging functions - in the LexiRes Editor.

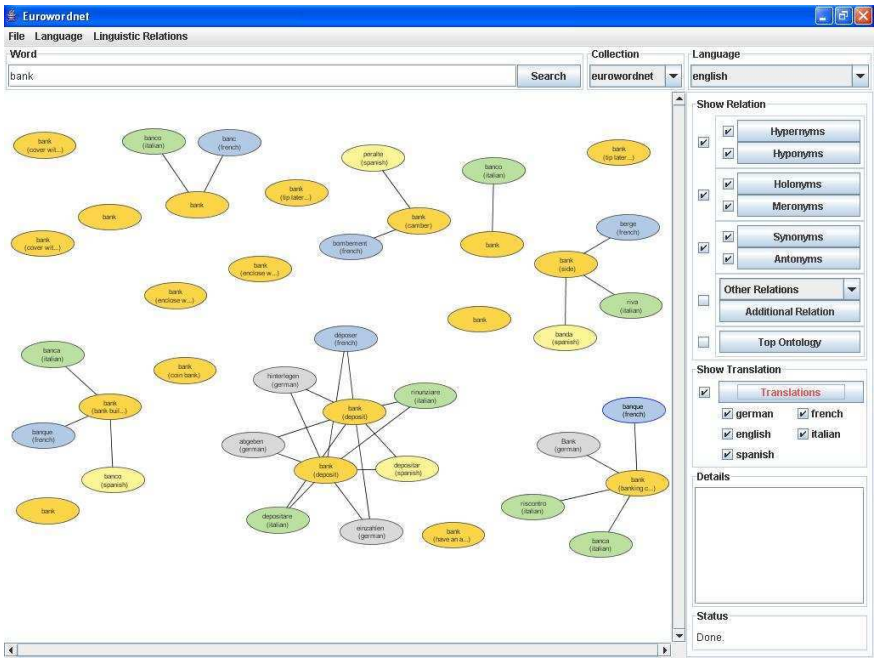


Figure 4. Example of the word ”bank” - SynSet translations - in the LexiRes Editor.

archical word specific overview that gives the possibility to restructure concepts using automatic or manual merging methods. These methods are important to obtain a lexical resource that is more appropriate in order to disambiguate user query words in documents retrieved from an information retrieval system.

REFERENCES

[1] Hork A. and Smr P., 'Visdic - wordnet browsing and editing tool.', in *Proceedings of the Second International WordNet Conference (GWC2004)*, (2004).

[2] Susan T. Dumais, Edward Cutrell, and Hao Chen, 'Optimizing search by showing results in context', in *CHI*, pp. 277–284, (2001).

[3] L. Bentivogli E. Pianta and C. Girardi., 'Multiwordnet: developing an aligned multilingual database.', in *First International Conference on Global WordNet*, Mysore, India, (2002).

[4] C. Fellbaum D. Gross G. Miller, R. Beckwith and K. Miller., 'Five papers on wordnet.', *International Journal of Lexicology*, **3(4)**, (1990).

[5] Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari, 'Conceptual analysis of lexical taxonomies: the case of wordnet top-level', in *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pp. 285–296, New York, NY, USA, (2001). ACM Press.

[6] N. Guarino and C. A. Welty., *An overview of OntoClean.*, 151–172, Handbook on Ontologies, Springer, 2004.

[7] Yannis Labrou and Timothy W. Finin, 'Yahoo! as an ontology: Using yahoo! categories to describe documents', in *CIKM*, pp. 180–187, (1999).

[8] Ernesto William De Luca and Andreas Nürnberger, 'Improving ontology-based sense folder classification of document collections with clustering methods', in *Proc. of 2nd Int. Workshop on Adaptive Multimedia Retrieval (AMR 2004), part of ECAI 2004*, eds., Philippe Joly, Marcin Detyniecki, and Andreas Nürnberger, (2004).

[9] Ernesto William De Luca and Andreas Nürnberger, 'Ontology-based semantic online classification of documents: Supporting users in searching the web', in *Proc. of the European Symposium on Intelligent Technologies (EUNITE 2004)*, (2004).

[10] Ernesto William De Luca and Andreas Nürnberger, 'Supporting mobile web search by ontology-based categorization', in *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen, Proc. of GLDV 2005*, eds., Bernhard Fisseni, Hans-Christian Schmitz, Bernhard Schröder, and Petra Wagner, pp. 28–41, (2005).

[11] Ernesto William De Luca and Andreas Nürnberger, 'The use of lexical resources for sense folder disambiguation.', in *Workshop Lexical Semantic Resources (DGfS-06)*, Bielefeld, Germany., (2006).

[12] E. Motta, S. Buckingham, and J. Domingue. Ontology-driven document enrichment: Principles and case studies, 1999.

[13] A. Oltramari, A. Gangemi, N. Guarino, and C. Masolo. Restructuring wordnet's top-level: The ontoclean approach.

[14] Pianta E. Ranieri M. and Bentivogli L., 'Browsing multilingual information with the multisemcor web interface', in *Proceedings of the LREC 2004 Satellite Workshop on The Amazing Utility of Parallel and Comparable Corpora*, pp. 38–41, Portugal, (2004).

[15] P. Vossen. Eurowordnet general document.

[16] David Yarowsky, 'Unsupervised word sense disambiguation rivaling supervised methods', in *Meeting of the Association for Computational Linguistics*, pp. 189–196, (1995).

Integrating tf-idf Weighting with Fuzzy View-Based Search

Markus Holi and **Eero Hyvönen** and **Petri Lindgren**[1]

Abstract. This paper presents a weighting method of document annotations for fuzzy view-based semantic search (FVBSS). FVBSS is a fuzzy generalization of semantic view-based search, which supports the ranking of search results according to relevance. The presented method is an extension of the *tf-idf* weighting method. Our approach takes into account the semantic relations between indexing terms and concepts which leads to more accurate and compressed represenatation of the document and flexible information retrieval. Our preliminary user evaluation indicates that the weighting method presented here produces document rankings that match human judgment in a promissing way.

1 INTRODUCTION

Semantic portals[2] [11] usually provide the user with two basic services: 1) A search engine based on the semantics of the content [5], and 2) dynamic linking between pages based on the semantic relations in the underlying knowledge base [6]. In this paper we concentrate on the first service, the semantic search engine.

One of the basic capabilities that are expected from a search engine is the ability to rank query results according to relevance. However, many otherwise competent semantic search engines — such as engines based on the view-based search paradigm [16, 7, 10] — do not provide this function. This follows from the fact that ontologies are based on crisp logic whereas ranking of results requires methods of uncertain reasoning.

To overcome this shortcoming we have created a fuzzy (FVBSS) generalization of the semantic view-based search paradigm, which is based on weighted document annotations [8]. FVBSS enables the ranking of search results according to relevance. This paper develops the paradigm further by presenting an automatic method to create fuzzy annotations from crisp ones. The fuzzy value reflects the relevance of the annotation to the document. The method is an ontological extension of the tf-idf [18] weighting method that is widely used in information retrieval systems.

The rest of the paper is organized as follows: In section 2 the semantic-view based search and its fuzzy extension will be described. In section 3 the ontological extension of the tf-idf weighting method will be presented. A test implementation and evaluation will be presented in section 4 and finally, section 5 summarizes the paper's contributions, discusses related work and presents learned lessons and directions for future research.

[1] Helsinki University of Technology (TKK), Media Technology and University of Helsinki, Finland, http://www.seco.tkk.fi/, email: firstname.lastname@tkk.fi

[2] See, e.g., http://www.ontoweb.org/ or http://www.semanticweb.org

2 VIEW-BASED SEARCH

2.1 Crisp View-Based Semantic Search

The view-based search paradigm[3] is based on *facet analysis* [13], a classification scheme introduced in information sciences by S. R. Ranganathan already in the 1930's. From the 1970's on, facet analysis has been applied in information retrieval research, too, as a basis for search. The idea of the scheme is to analyze and index search items along multiple orthogonal taxonomies that are called subject *facets* or *views*. Subject headings can then be synthesized based on the analysis. This is more flexible than the traditional library classification approach of using a monolithic subject heading taxonomy.

In view-based search [16, 7, 10], the views are exposed to the end-user in order to provide her with the right query vocabulary, and for presenting the repository contents and search results along different views. The query is formulated by constraining the result set in the following way: When the user selects a category c_1 in a view v_1, the system constrains the search by leaving in the result set only such objects that are annotated (indexed) in view v_1 with c_1 or some subcategory of it. When an additional selection for a category c_2 from another view v_2 is made, the result is the intersection of the items in the selected categories, i.e., $c_1 \cap c_2$. After the result set is calculated, it can be presented to the end-user according to the view hierarchies for better readability. This is in contrast with traditional search where results are typically presented as a list of decreasing relevance.

View-based search has been integrated with the notion of ontologies and the semantic web [10, 15, 9, 12]. The idea of such *semantic view-based search* is to construct facets algorithmically from a set of underlying ontologies that are used as the basis for annotating search items. Furthermore, the mapping of search items onto search facets can be defined using logic rules. This facilitates more intelligent "semantic" search of indirectly related items. Another benefit is that the logic layer of rules makes it possible to use the same search engine for content annotated using different annotation schemes. Ontologies and logic also facilitates *semantic browsing*, i.e., linking of search items in a meaningful way to other content not necessarily present in the search result set.

2.2 Fuzzy Semantic View-Based Search

The view-based search scheme has also some shortcomings. First, it does not incorporate the notion of relevance. Thus, there is not a way to rank the results according to relevance. Second, in semantic view-based search the views are generated according to the concept

[3] A short history of the parading is presented in http://www.view-based-systems.com/history.asp

hierarchies of the ontologies. This is not always ideal from the viewpoint of the end user, because the ontologies usually are created by and for domain experts, and thus it contains concepts and structures that might not be familiar or intuitive to a non-professional end-user.

To overcome these problems we created a fuzzy version of the search paradigm [**?**]. In this paradigm 1) the degrees of relevance of documents can be determined and 2) distinct end-user's views to search items can be created and mapped onto indexing ontologies and the underlying search items (documents). The framework generalizes view-based search from using crisp sets to fuzzy set theory and is called *fuzzy view-based semantic search*.

2.2.1 The Architecture of the Framework

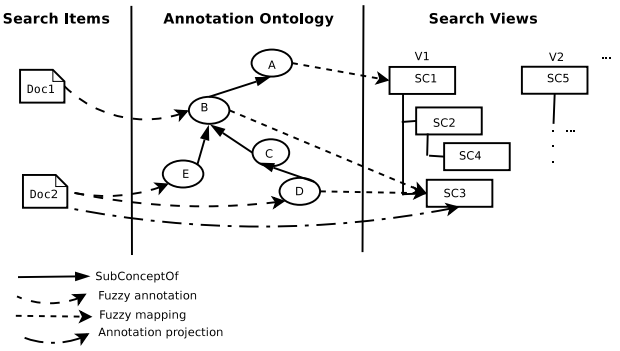


Figure 1. Components of the fuzzy view-based semantic framework

The architecture of the framework is depicted in figure 1. The framework consists of the following components:

Search Items The search items are a finite set of documents D depicted on the left. D is the fundamental set of the fuzzy view-based search framework.

Annotation Ontology The search items are annotated according to the ontology by the indexer. The ontology consists of two parts. First, a finite set of annotation concepts AC, i.e. a set of fuzzy subsets of D. Annotation concepts $AC_i \in AC$ are atomic. Second, a finite set of annotation concept inclusion axioms $AC_i \subseteq AC_j$[4], where $AC_i, AC_j \in AC$ are annotation concepts and $i, j \in N$, and $i \neq j$. These inclusion axioms denote subsumption between the concepts and they constitute a concept hierarchy.

Search Views Search views are hierarchically organized search categories for the end-user to use during searching. The views are created and organized with end-user interaction in mind, and may not be identical to the annotation concepts for professional indexers. Each search category SC_i is a fuzzy subset of D. In crisp view-based search the intersection of documents related to selected search categories is returned as the result set, while in fuzzy view-based search, the intersection is replaced by the fuzzy intersection.

Search items related to a search category SC_i can be found by mapping them first onto annotation concepts by annotations, and then by mapping annotation concepts to SC_i. The result R is not a crisp set of search items $R = SC_1 \cap ... \cap SC_n = \{Doc_1, ..., Doc_m\}$ as in view-based search, but a fuzzy set where the relevance of each item is specified by the value of the membership function mapping:

$$R = SC_1 \cap ... \cap SC_n = \{(Doc_1, \mu_1), ..., (Doc, \mu_m)\}$$

In the following the required mappings are described. For a fuller description see [**?**].

2.2.2 Fuzzy Annotations

Search items (documents) have to be annotated in terms of the annotation concepts—either manually or automatically by using e.g. logic rules. In (semantic) view-based search, the annotation of a search item is the crisp set of annotation concept categories in which the item belongs to. In figure 1, annotations are represented using bending dashed arcs from *Search Items* to *Annotation Ontology*. For example, the annotation of the item *Doc2* would be the set $A_{Doc2} = \{E, D\}$.

In our approach, the relevance of different annotation concepts with respect to a document may vary and is represented by a *fuzzy annotation*. The fuzzy annotation A_D of a document D is the set of its fuzzy concept membership assertions:

$$A_D = \{(AC_1, \mu_1), ..., (AC_n, \mu_n)\} \text{ where } \mu_i \in (0, 1]$$

Here μ_i tell the degrees by which the annotated document is related to annotation concepts AC_i. For example, our test dataset consisted of health related documents that were annotated using the finnish translation of Medical Subject Headings (MeSH)[5]. A document D_1 from that document set was given the fuzzy annotation

$$A_{D_1} = \{(Exercise, 0.3), (Diet, 0.4)\}$$

Based on the annotations, the membership function of each fuzzy set $AC_j \in AC$ can be defined. This is done based on the meaning of subsumption, i.e. inclusion. One concept is subsumed by the other if and only if all individuals in the set denoting the subconcept are also in the set denoting the superconcept, i.e., if being in the subconcept implies being in the superconcept [17]. Thus, applying this principle to fuzzy sets we define the membership degree of a document D_i in AC_j as the maximum of its concept membership assertions made for the subconcepts of AC_j.

$$\forall D_i \in D, \mu_{AC_j}(D_i) = max(\mu_{AC_i}(D_i))$$

where $AC_i \subseteq AC_j$.

For example, assume that we have a document D_2 that is annotated with the annotation concept *Asthma* with weight 0.8, i.e. $\mu_{Asthma}(D_2) = 0.8$. Assume further, that in the annotation ontology *Asthma* is a subconcept of *Diseases*, i.e. $Asthma \subseteq Diseases$. Then,

$$\mu_{Diseases}(D_2) = \mu_{Asthma}(D_2) = 0.8$$

2.2.3 Fuzzy Mappings

Each search category SC_i in a view V_j is defined using concepts from the annotation ontology by a finite set of fuzzy concept inclusion axioms that we call *fuzzy mappings*:

[4] Subset relation between fuzzy sets is defined as: $AC_i \subseteq AC_j$ iff $\mu_{AC_j}(D_i) \geq \mu_{AC_i}(D_i), \forall D_i \in D$, where D is the fundamental set.

[5] http://http://www.nlm.nih.gov/mesh/

$$AC_i \subseteq_\mu SC_j \quad \text{where } AC_i \in AC, SC_j \in V_k,\ i,j,k \in N \text{ and } \mu \in (0,1]$$

A fuzzy mapping describes the meaning of a search category SC_j by telling to what degree μ the membership of a document D_i in an annotation concept AC_i implies its membership in SC_j. Intuitively, a fuzzy mapping reveals to which degree the annotation concept can be considered a subconcept of the search category. In figure 1, fuzzy mappings are represented using straight dashed arcs.

Thus, fuzzy inclusion is interpreted as fuzzy implication. The definition is based on the connection between inclusion and implication described previously. This is extended to fuzzy inclusion as in [20, 4]. We use Goguen's fuzzy implication, i.e.

$$i(\mu_{AC_j}(D_i), \mu_{SC_i}(D_i)) = 1 \text{ if } \mu_{SC_i}(D_i) \geq \mu_{AC_j}(D_i) \text{ else } \mu_{SC_i}(D_i)/\mu_{AC_j}(D_i) \qquad \forall D_i \in D$$

Let us continue with the example case in the end of section 2.2.2 where we defined the membership of document D_1 in the annotation concept $Diseases$. Assume that we want to define a search category *Food and Diseases* that will give the user information about the relation of diseases to food.

As part of the category definition we would create a fuzzy mapping

$$Diseases \subseteq_{0.1} \textit{Food and Diseases}$$

Based on this fuzzy mapping, the membership degree of the document $D1$ in *Food and Diseases* is

$$\mu_{\textit{Food and Diseases}}(D1) = \mu_{Diseases}(D1) * 0.1 = 0.8 * 0.1 = 0.08$$

A search category SC_j is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Fuzzy mappings can be created by a human expert or by an automatic or a semi-automatic ontology mapping tool.

Mappings can be nested. Two fuzzy mappings $M_1 = AC_i \subseteq_\mu SC_i$ and $M_2 = AC_j \subseteq_\nu SC_i$ are *nested* if $AC_i \subseteq AC_j$, i.e., if they point to the same search category, and one of the involved annotation concepts is the subconcept of the other. In this case the more specific mapping is relevant for a document when computing its membership in the search category SC_j.

It is also possible to map a search category to a Boolean combination of annotation concepts. In this cases the membership functions of these boolean concepts are calculated according to the fuzzy union, intersection and negation operations.

2.2.4 *Performing the Search*

In view-based search the user can query by choosing concepts from the views. In crisp semantic view-based search, the extension E of a search category is the union of its projection P and the extensions of its subcategories S_i, i.e. $E = P \bigcup S_i$. The result set R to the query is simply the intersection of the extensions of the selected search categories $R = \bigcap E_i$ [9].

In fuzzy view-based search we extend the crisp union and intersection operations to fuzzy intersection and fuzzy union. Recall, from section 2.2.3 that a search category was defined as the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Thus, the fuzzy union part of the view-based search is already taken care of. Now, if E is the set of selected search categories, then the fuzzy result set R is the fuzzy intersection of the members of E, i.e. $R = SC_1 \cap ... \cap SC_n$, where $SC_i \in E$.

Using Gödel's intersection [24], we have:

$$\mu_R(D_k) = min(\mu_{SC_1}(D_k), ..., \mu_{SC_n}(D_k)) \forall D_k \in D$$

As a result, the answer set R can be sorted according to relevance in a well-defined manner, based on the values of the membership function.

3 AUTOMATIC CREATION OF FUZZY ANNOTATIONS

We created the fuzzy annotations with an ontological extension of the tf-idf weighting method. In the following first the tf-idf weighting method is described and then our ontological extension of it is presented.

3.1 Tf-idf

The tf-idf [18] (term frequency - inverse document fequency) weighting method is often used in information retrieval. It is a statistical technique to evaluate how important a term is to a document. The importance increases proportionally to the number of times a word appears in the document but is offset by how common the word is in all of the documents in the document collection. Tf-idf is often used by search engines to find the most relevant documents to a user's query. There are many different formulas used to calculate tf-idf. A widely used formula that calculates a normalized tf-idf weight[6] is presented below. The formula gives values between 0 and 1.

The term frequency $tf_{t_i Doc_j}$ of term t_i in a document Doc_j gives a measure of the importance of the term within the document. In the formula that we used $tf_{t_i Doc_j}$ is simply the number of occurrences of t_i in Doc_j.

The inverse document frequency idf is a measure of the general importance of the term. In the formula that we used idf_{t_i} is the natural logarithm of the number of all documents N divided by df_{t_i} — the number of documents containing the term t_i, i.e.

$$idf_{t_i} = log(\frac{N}{df_{t_i}})$$

The normalized tf-idf weight of the term t_i in document Doc_j is

$$\textit{tf-idf}_{t_i Doc_j} = \frac{tf_{t_i Doc_j} * idf_{t_i}}{\sqrt{\sum_{i=1}^{M} (tf_{t_i Doc_j} * idf_{t_i})^2}}$$

where M is the number of terms in Doc_j. A high weight in tf-idf is reached by a high term frequency in the given document and a low document frequency in the whole collection of documents.

3.2 Ontological Extension of tf-idf

We extended the tf-idf weighting method so that it can be used to weight existing crisp document annotations. Recall from chapter 2.2.2 that a crisp annotation of a document Doc_j is the set $A_{Doc_j} = \{AC_1, ..., AC_n\}$, where $AC_1, ..., AC_n$ are concepts of

[6] See, http://www.sims.berkeley.edu:8000/courses/is202/f05/LectureNotes/202-20051110.pdf

the annotation ontology. The weighting is done based on the textual content of the document and the description of each concept $AC_1, ..., AC_n$ in the ontology.

The main idea is that instead of calculating the importance of each word to a given document Doc_j we calculate the importance of each concept in A_{Doc_j} to the document. The weighting of the annotation of Doc_j is done as follows:

1. A set of words W_{AC_i} is created for each concept in A_{Doc_j}. The set is the union of the labels of AC_i and the labels of the subconcepts of AC_i in the ontology.
2. The term frequency $tf_{AC_iDoc_j}$ for each AC_i in Doc_j is counted. This is done by reading (automatically) through Doc_j and each time that a word that belongs to the set W_{AC_i} is encountered $tf_{AC_iDoc_j}$ is increased by one. The counter starts from 1, thus if there are no occurrences of AC_i in Doc_j, then $tf_{AC_iDoc_j} = 1$. This is to recognize the fact that if a document is annotated using AC_i then AC_i is relevant to the document even if the content does not speak of AC_i directly.
3. The number of documents annotated with AC_i, i.e. df_{AC_i} is counted.

Now

$$idf_{AC_i} = log(\frac{N}{df_{AC_i}})$$

where N is the number of documents in the collection, and

$$\textit{tf-idf}_{AC_iDoc_j} = \frac{tf_{AC_iDoc_j} * idf_{AC_i}}{\sqrt{\sum_{i=1}^{M}(tf_{AC_iDoc_j} * idf_{AC_i})^2}}$$

where M is the number of concepts in A_{Doc_j}.

The ontological extension of tf-idf presented above offers some benefits when compared to traditional tf-idf. The benefits are a result of the utilization of the structure of the annotation ontology. First, terms that are expressions of the same concept are detected. Thus they can be represented using a single concept identifier and the representation of the document content is compressed. Second, the concept hierarchies enable a better query answering. For example, the system knows that documents about dogs are relevant to a query about animals.

4 TEST IMPLEMENTATION AND EVALUATION

We implemented the representation of ontologies, annotations and search views using RDF [1]. The algorithms were implemented using Java[7] and its Semantic Web Framework Jena[8]. Next we will describe the document collection, the ontology and the search views of our test implementation and then a preliminary evaluation of the method will be presented.

4.1 Document Collection and Ontology

Our document set consisted of 163 documents from the web site of the National Public Health Institute[9] of Finland (NPHI).

As an annotation ontology we created a SKOS [2] version of FinMeSH, the Finnish translation of MeSH. The fuzzy annotations were created in two steps. First, an information scientist working for the NPHI annotated each document with a number of FinMeSH concepts. These annotations were crisp. Second, the crisp annotations were weighted using an ontological extension of tf-idf described above. The search views with the mappings were designed and created by hand.

4.2 Evaluation

The main practical contribution of our framework in comparison to crisp view-based search is the ranking of search results according to relevance. A preliminary user-test was conducted to evaluate the ranking done by the implementation described above. The test group consisted of five subjects.

The test data was created in the following way. Five search categories were chosen randomly. These categories were: Diabetes, Food, Food Related Diseases, Food Related Allergies, and Weight Control. The document set of each category was divided into two parts. The first part consisted of the documents who's rank was equal or better than the median rank, and the second part consisted of documents below the median rank. Then a document was chosen from each part randomly. Thus, each of the chosen categories was attached with two documents, one representing a well ranking document, and the other representing a poorly ranking document.

The test users were asked to read the two documents attached to a search category, e.g. Diabetes, in a random order, and pick the one that they thought was more relevant to the search category. This was repeated for all the selected search categories. Thus, each tested person read 10 documents.

The relevance assessment of the test subjects were compared to the ordering done by our implementation. According to the results every test subject ordered the documents in the same way that the algorithm did.

5 DISCUSSION

This paper presented an ontological extension to the widely used tf-idf weighting method. It was designed to enable the automatic weighting of crisp document annotations based on the textual content of each document and the conceptual information of the ontology. The ontological tf-idf method evaluates the importance of the annotation concept to the document.

5.1 Contributions

The main benefits of the method when compared to the traditional tf-idf method are: First, terms that are expressions of the same concept can be represented using a single concept identifier which results in a compressed representation of the document content. Second, the concept hierarchies of the ontologies can be utilized to enable better query answering.

We integrated the method to our FVBSS framework as a way to automatically create fuzzy annotations from crisp annotations. FVBSS enables the ranking of search results according to query relevance in view-based semantic search. A prototype implementation and its application to a data set in semantic eHealth portal was discussed and evaluated.

5.2 Related Work

The fuzzy semantic view-based search framework presented in this paper generalizes the traditional view-based search paradigm [16, 7,

[7] http://java.sun.com
[8] http://jena.sourceforge.net
[9] http://www.ktl.fi/

10] and its semantic extension developed in [10, 15, 9, 12]. Fuzzy reasoning [22] has been used before in IR but to our knowledge not with (semantic) view-based search.

We have applied the idea presented by Straccia [20] in his fuzzy extension to the description logic *SHOIN(D)* and Bordogna [4] of using fuzzy implication to model fuzzy inclusion between fuzzy sets. Also other fuzzy extensions to description logic exist, such as [19, 14].

Zhang et al. [23] have applied fuzzy description logic and information retrieval mechanisms to enhance query answering in semantic portals. Their framework is similar to ours in that both the textual content of the documents and the semantic metadata is used to improve information retrieval. However, the main difference in the approaches is that their work does not help the user in query construction whereas the work presented in this paper does by providing an end-user specific view to the search items.

Akrivas et al. [3] present an interesting method for context sensitive semantic query expansion. In this method, user's query words are expanded using fuzzy concept hierarchies. An inclusion relation defines the hierarchy. The inclusion relation is defined as the composition of subclass and part-of relations. Each word in a query is expanded by all the concepts that are included in it according to the fuzzy hierarchy.

In [3], the inclusion relation is of the form $P(a, b) \in [0, 1]$ with the following meaning: A concept a is completely a part of b. High values of the $P(a, b)$ function mean that the meaning of a approaches the meaning of b. Thus, the difference to our work is that the inclusion relation in in the fuzzy hierarchy of [3] is crisp, whereas in our *fuzzy mappings* the inclusion relation itself is fuzzy.

Widyantoro and Yen [21] have created a domain-specific search engine called PASS. The system includes an interactive query refinement mechanism to help to find the most appropriate query terms. The system uses a fuzzy ontology of term associations as one of the sources of its knowledge to suggest alternative query terms. The ontology is organized according to narrower-term relations. The ontology is automatically built using information obtained from the system's document collections. The fuzzy ontology of Widyantoro and Yen is based on a set of documents, and works on that document set. The automatic creation of ontologies is an interesting issue by itself, but it is not considered in our paper. At the moment, better and richer ontologies can be built by domain specialists than by automated methods.

5.3 Lessons Learned and Future Work

The ontological extension of the tf-idf weighting method proved to be rather straight forward to design and implement. Our preliminary evaluation of ranking search results with the framework were promising. However, the number of test subjects and the size of test data set was still too small for proper statistical analysis.

Our framework did get some inspiration from fuzzy versions of description logics. We share the idea of generalizing the set theoretic basis of an IR-system to fuzzy sets in order to enable the handling of vagueness and uncertainty. In addition, the use of fuzzy implication to reason about fuzzy inclusion between concepts is introduced in the fuzzy version [20] of the description logic *SHOIN(D)*. However, the ontologies that we use are mainly simple concept taxonomies, and in many practical cases we saw it as an unnecessary overhead to anchor our framework in description logics.

Furthermore, the datasets in our *Tervesuomi.fi* eHealth portal case study are large. The number of search-items will be probably between 50,000 and 100,000, and the number of annotation concepts probably between 40,000 and 50,000. For this reason we wanted to build our framework on the view-based search paradigm that has proven to be scalable to relatively large data sets. For example, the semantic view-based search engine *OntoViews* was tested to scale up to 2.3 million search items and 275,000 search categories in [12]. The fuzzy generalization adds only a constant coefficient to the computational complexity of the paradigm.

In the future we intend to implement the framework with a larger dataset in the semantic *Tervesuomi.fi* eHealth portal and test it with a larger user group. The fuzzy framework will be attached to the *OntoViews* tool as a separate ranking module. Thus, there is not a need for major refactoring of the search engine in *OntoViews*.

In addition we intend to apply the framework to the ranking of the recommendation links created by *OntoDella*, which is the semantic recommendation service module of *OntoViews*.

ACKNOWLEDGEMENTS

Our research was funded mainly by the National Technology Agency Tekes. The National Public Health Institute of Finland (NPHI) provided us with the data annotated by Johanna Eerola.

REFERENCES

[1] *RDF Primer*. http://www.w3.org/TR/rdf-primer.

[2] *SKOS Core Guide*, 2005. http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.

[3] G. Akrivas, M. Wallace, G. Andreou, G. Stamou, and S. Kollias, 'Context - sensitive semantic query expansion', in *Proceedings of the IEEE International Conferrence on Artificial Intelligence Systems (ICAIS)*, (2002).

[4] G. Bordogna, P. Bosc, and G. Pasi, 'Fuzzy inclusion in database and information retrieval query interpretation', in *ACM Computing Week - SAC'96*, Philadelphia, USA, (1996).

[5] S. Decker, M. Erdmann, D. Fensel, and R. Studer, 'Ontobroker: Ontology based access to distributed and semi-stuctured information', *DS-8*, 351–369, (1999). http://citeseer.nj.nec.com/article/decker98ontobroker.html.

[6] C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall, 'Conceptual open hypermedia = the semantic web', in *Proceedings of the WWW2001, Semantic Web Workshop*, Hongkong, (2001).

[7] M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Lee, 'Finding the flow in web site search', *CACM*, **45**(9), 42–49, (2002).

[8] Markus Holi and Eero Hyvönen, 'Fuzzy view-based semantic search', in *Proceedings of the Asian Semantic Web Conference (ASWC2006)*, (September 2006). To be published.

[9] Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula, 'Museumfinland – finnish museums on the semantic web', *Journal of Web Semantics*, **3**(2), 25, (2005).

[10] Eero Hyvönen, Samppa Saarela, and Kim Viljanen, 'Application of ontology techniques to view-based semantic search and browsing', in *The Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium (ESWS 2004)*, (2004).

[11] A. Maedche, S. Staab, N. Stojanovic, R. Struder, and Y. Sure, 'Semantic portal - the seal approach', Technical report, Institute AIFB, University of Karlsruhe, Germany, (2001).

[12] Eetu Makelä, Eero Hyvönen, Samppa Saarela, and Kim Viljanen, 'Ontoviews – a tool for creating semantic web portals', in *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, (May 2004).

[13] A. Maple. Faceted access: a review of the literature, 1995. http://library.music.indiana.edu/tech_s/mla/facacc.rev.

[14] M. Mazzieri and A. F. Dragoni, 'Fuzzy semantics for semantic web languages', in *Proceedings of ISWC-2005 Workshop Uncertainty Reasoning for the Semantic Web*, (Nov 2005).

[15] Eetu Mäkelä, Eero Hyvönen, and Teemu Sidoroff, 'View-based user interfaces for information retrieval on the semantic web', in *Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction*, (Nov 2005).
[16] A. S. Pollitt, 'The key role of classification and indexing in view-based searching', Technical report, University of Huddersfield, UK, (1998). http://www.ifla.org/IV/ifla63/63polst.pdf.
[17] A. Rector, 'Defaults, context, and knowledge: Alternatives for owl-indexed knowledge bases', in *Proceedings of Pacific Symposium on Biocomputing*, (2004).
[18] G. Salton and C. Buckley, 'Term weighting approaches in automatic text retrieval', Technical report, Ithaca, NY, USA, (1987).
[19] G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, and I. Horrocks, 'The fuzzy description logic f-shin', in *Proceedings of ISWC-2005 Workshop Uncertainty Reasoning for the Semantic Web*, (Nov 2005).
[20] Umberto Straccia, 'Towards a fuzzy description logic for the semantic web (preliminary report)', in *2nd European Semantic Web Conference (ESWC-05)*, number 3532 in Lecture Notes in Computer Science, pp. 167–181, Crete, (2005). Springer Verlag.
[21] D.H. Widyantoro and J. Yen, 'A fuzzy ontology-based abstract seawrch engine and its user studies', in *The Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, (2002).
[22] L. Zadeh, 'Fuzzy sets', *Information and Control*, (1965).
[23] L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang, 'An enhanced model for searching in semantic portals', in *Proceedings of the Fourteenth International World Wide Web Conference*, (May 2005).
[24] H.-J. Zimmermann, *Fuzzy Set Theory and its Applications*, Springer, 2001.

Framework for Semi Automatically Generating Topic Maps

Lóránd Kásler[1] and **Zsolt Venczel**[1] and **László Zsolt Varga**[1]

Abstract. The amount of electronically stored textual information is continuously increasing both on the internet and in company assets, and there are no good solutions to easily locate the most needed information. Because search engines do not take into account the meaning of the word and its context, in the end the user has to select the right information from the unstructured result set. If the text is annotated and linked to the ontology of the annotation, then the user can directly navigate along the links of the semantic annotation to the desired information.

In this paper we present a software framework to semi automatically generate a semantic representation of the knowledge of the Networkshop conference series and display on a web portal the generated ontology together with the references to the occurrences of the instances in the source text. The framework presented in this paper makes advances in the following fields: we do not assume that the source text has uniform and formally defined structure, we address English and Hungarian text as well, we incorporate machine learning techniques in the process, and provide a flexible content management system for the presentation of the generated Topic Map on a web based portal.

1 INTRODUCTION

The amount of electronically stored textual information is continuously increasing both on the internet and in company assets, and there are no good solutions to easily locate the most relevant information. Although there are engines, like Google, for word based search, and the techniques are continuously improved, in the end the real search is completed by the user, because these search engines do not take into account the meaning of the word and its context. The semantic web tries to improve this by attaching semantic annotation to data and text. The semantic annotation is based on an ontology of the given domain. However the amount of information is huge and the semantic annotation cannot be done manually for large amount of text, therefore there is need for automated tools.

Once the information is semantically annotated, then the search can be improved in two ways. One way is that the search engine takes into account the semantic annotation of the text in order to further improve the result set of the search; the other is that there is no search engine and the user directly navigates along the links of the ontology of the semantic annotation to the desired information. The second approach has limitations towards large information sets like the whole internet, however in the scales of single portals or company information assets this can be a viable option. In addition the second approach has advantages as well. One advantage is that the ontology of the semantic annotation is closer to the thinking of the user and the user feels it much more comfortable to browse along the ontology than to select the right information from the unstructured result set of search engines. This holds in our case where we build a portal and a knowledge source specialised on a specific domain. Another advantage is that while navigating along the ontology of the annotation, the user may find other relevant and interesting information which he/she would not even think of.

Currently there are two main standards for representing the knowledge used for the annotation: the W3C standard [5] RDF/OWL and the ISO standard Topic Map [6]. We chose the Topic Map standard, because its concept is like and intelligent extension of the index of books with key features as topics, associations between topics, and occurrences of topics. We also chose the Topic Map standard, because it is very flexible in merging and extending different sets of Topic Maps.

In this paper we present a software framework to semi automatically generate a semantic representation of the knowledge and information present in a set of natural language text files, and display the generated ontology together with the reference to their occurrences in the source text on a web portal. The software framework is applied to semi automatically generate a Topic Map from ten years of the NetWorkshop conference proceedings [38]. The result is presented on a structured information portal and content management system.

The development of this software framework was motivated by the challenge of applying the Topic Map technology, the lack of such a framework, the lack of specialized and fast algorithm implementations with high precision on a medium data corpus. The implementation takes into account the specialities of the Hungarian language, mainly the problems of stemming.

The specific task, to semi-automate the construction of a Topic Map based on a conference was originally tackled by Steve Pepper and Lars Marius Garshol from Ontopia [1]. Their original intent was to describe a showcase on applying Topic Maps on real data rather than experimenting text mining algorithms and heuristics. The abstract concept of generating a Topic Map from any kind of semi-structured data is still an open field. Concrete techniques are mentioned in [2], or implemented in TMHarvest [3]. The Topic Map generating framework presented in this paper is an independent development from the above works and makes advancements in the following fields: we do not assume that the source text has uniform and formally defined structure, we address English and Hungarian text as well, we incorporate machine

[1] Computer and Automation Research Institute, Kende u. 13-17., Budapest, 1111 Hungary email:laszlo.varga@sztaki.hu

learning and information retrieval [18][19] techniques in the process, and provide a flexible content management system for the presentation of the generated Topic Map on a web based portal.

The structure of the paper is as follows: in Section 2 we summarize the technology we build on, in Section 3 we describe the framework that we developed for generating Topic Maps, in Section 4 we evaluate the framework and the generated Topic Map portal.

2 APPLIED TECHNOLOGIES

In this section we are going to summarize the technologies used for the development of the software framework. We used a broad spectrum of mature, open-source, Java technologies.

2.1 Topic Map

Topic Maps became an ISO standard in January 2000 [6], and the technology is in active development. Considered by some a rival, or a redundant specification for the WWC standard RDF [5], but the two specifications address different needs [9][10][17], and can coexist in several ways [4][8].

The key features of Topic Maps are: topics identified by their names; associations between topics; and occurrences of topics pointed to via locators. The key main advantages of this knowledge representation technology are data merging, Published Identities [14], rich set of metadata, and an element named "scope", which is mainly used for multilingual purposes [11][12][13].

There are many Open Source [33][34] and commercial implementations of Topic Map in Java, from Ontopia, Infoloom, Empolis and other vendors. There is even an effort to standardize the API used by vendors, called Topic Map API (TMAPI) [15].

2.2 Machine Learning in Java

For various analysis tasks the framework uses several machine learning and language processing techniques [20]. One of the most comprehensive architecture and collection of algorithms in this field is an open-source project of the University of Waikato, named Weka Machine Learning Project [21]. Besides broad variety of implemented classifiers, there are other, open-source extensions like jBNC [29].

Among several advanced algorithms, the framework contains a pluggable stemming package. We have successfully integrated a Hungarian language stemming software package, called Szószablya [27]. This way all other layers of the application dependent on stemming became language independent, because the abstract stemming package instantiates the needed sub package.

Although WEKA is one of the popular choices for machine learning and text mining tasks, we experimented with other frameworks such as YALE (Yet Another Learning Environment) [28] as well.

2.3 Ant Framework

The Ant Framework [16] is known as an open source build system, but besides being a modern replacement for make, its task oriented philosophy, easy configuration and integrated command line interface has a larger applicability. The main phases of the process implemented in our framework are modeled as Ant Tasks and can be controlled uniformly.

3 FRAMEWORK FOR GENERATING TOPIC MAPS

The framework for generating topic maps consists of a set of software tools and methods to support the execution of the process represented on Figure 1. The process has four phases: the data organisation, the analysis, the Topic Map population and the content management phase.

In the data organisation phase the raw source text available in various formats and structures is processed to have uniform structure. In this phase the metadata that can be extracted from the semi structure of the raw text is extracted and converted to a formal structure.

The goal of the analysis phase is to identify the main topics and their associations present in the source text. Two basic identification methods are applied. One is the identification of the topics and associations from the structure of the source text. For example topics like the paper title, the author, the affiliation of the author can be identified by pointing to the appropriate item in the structured metadata. We did not use named entity recognition, because we could not have defined associations between recognised entities easily. Associations like "a paper is authored by an author" can be identified by associating the items in the structured metadata. The other identification method is based on the analysis of the natural language text of the source text. Ideally this could be based on information retrieval methods to identify the topics and their associations mentioned in the papers. The implementation of the method on natural language understanding would have been too ambitious for our project, therefore we decided to use already existing external taxonomy or ontology to assign keywords to papers. The associations between keywords are defined by the external ontology. The result of the analysis phase is a Topic Map skeleton which is a combination of the external ontology and the ontology defined by the source text structure.

The Topic Map skeleton contains topic types which do not have occurrences. For example we know that there are authors and papers, the authors can write papers on different keywords, "Java Virtual Machine" and "operating system" are keywords, Windows and Linux are operating systems, and Java Virtual Machines can have implementations on different operating systems. However we do not know which authors wrote about which keyword and we do not know which papers contain which keyword. In the Topic Map population phase we identify the concrete instances of these topic types identified in the analysis phase. The result of the Topic Map population phase is a complete Topic Map of the source text.

The final phase of the framework is the content management phase. In this phase the completed Topic Map is loaded into an informational portal where the Topic Map can be presented to the user in a user friendly way using a content management system. With the help of the content management system the screen of the portal can be formatted and transitive associations can be added. For example if we know that authors write papers and papers are about keywords, then we can add the transitive association that authors write about keywords.

In the following we are going to detail the phases of the process of generating Topic Maps.

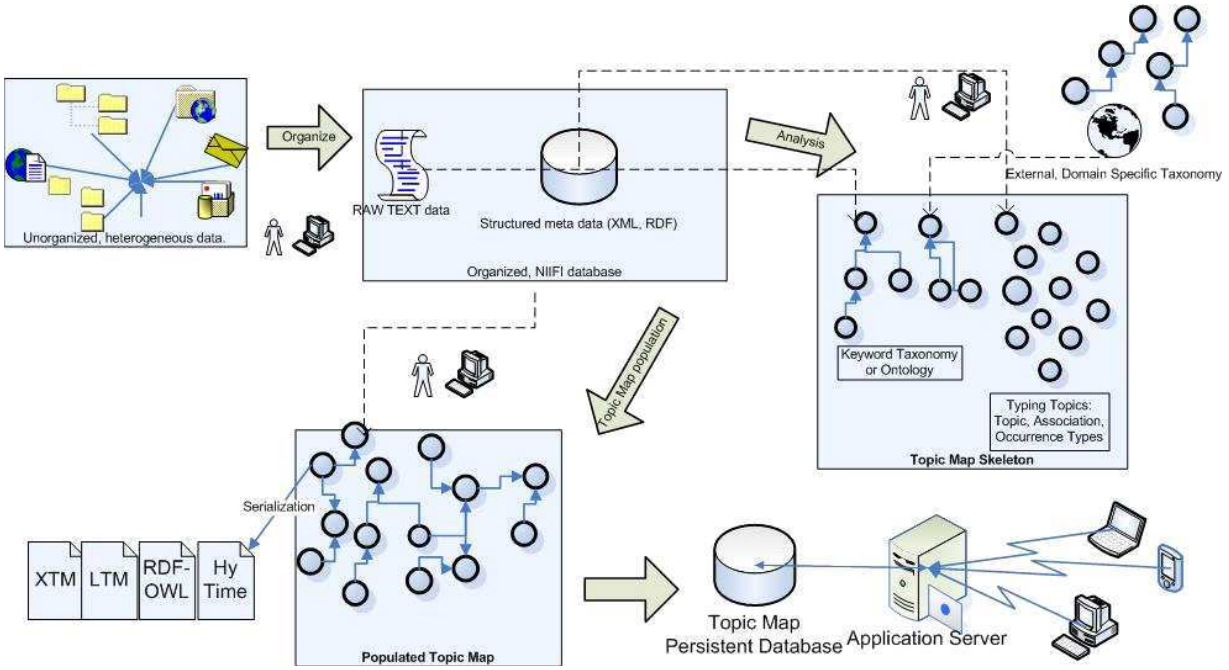


Figure 1. Framework for Generating Topic Maps

3.1 Phases of the Framework

Our solution consists of a semi-automatic system, capable of generating Topic Maps, from arbitrary complex data. It is a collection of tools, implemented in Java, forged together by a command line interface. The main design considerations were versatility, pluggability, runtime efficiency and incremental build. The project secondary objective, besides running a public portal, based on Topic Map technology, was to implement a Content Management backend for it. The backend leverages the used knowledge representation, thus it also has a Graphical Web interface for modifying the Topic Map. In order to achieve this, we had to maintain the incremental aspect of generating Topic Maps along all the tools.

The main task of generating the Topic Map is distributed over several runtime phases: Data Organization, Analysis, Topic Map Population. The Content Management Part of the Informational Portal based on the constructed Topic Map model, is in fact using the same architecture, to incrementally change underlying data. The generated Topic Map is persisted in a relational database, or in XTM [7], the XML interchange format for Topic Maps. The framework itself is agnostic of the chosen persistence alternatives, or the Topic Map engine, because it uses an abstract and standard API, called TMAPI.

As mentioned above these phases are incremental, which means, that they can augment any Topic Map model, indifferent of the used Topic Map engine and persistence technique. Most of the phases are semi-automatic, which means that user interaction is required to configure parameters or confirm certain assumptions made by several heuristic algorithms.

3.1.1 Data Organization Phase

Originally the data from the ten Networkshop conferences were in various formats and scattered in different places. Almost every conference had a different structure, or even worse: similar, but randomly discrepant directory trees. In this phase, we collected all the metadata from the data corpus and stored it in a structured way, using XML. The metadata to be extracted is identified by looking at the format of the papers and identifying for example that the first line is the title, the second is the authors list, etc. Several pattern matching techniques are used, such as regular expressions, to construct this preliminary database. The tools are manually configured to gather as much useful information as possible.

Another important part is textual data extractions from the different file formats used, because the conference paper formats changed from year to year. The parsers used are also Open Source implementations of parsers for the popular formats, such as Microsoft Word doc, PowerPoint, pdf and others.

3.1.2 Analysis Phase

As we said previously, the process of analysis leads to a Topic Map skeleton, containing the Typing Topics and Keyword topic instances.

The collection of typing topics in a Topic Map represents the ontology used by that model. Every topic is an instance of one of these typing topics. This ontology is the core, on which other tasks depend and it constitutes a solid base, on which layers of concrete data can be built. The process of discovering Typing Topics is also manually configured. Basically for every structured metadata

format the user has to create an XML configuration file containing the mappings. We used and enhanced the TMHarvest framework for this task. The mapping file contains several patterns, like XPath expressions, or Regular expressions to encapsulate the source of a typing topic. For the current data corpus we identified eleven typing topics, like Paper, Author, Conference and also other Association and Occurrence types.

The keyword topics are taken from an external ontology which may come from several sources. The actual implementation uses an external source, FOLDOC [24] to obtain a rich, domain specific ontology. Other taxonomies, web directories, or dictionaries could be easily used, such as ODP [25] and Babel [26]. The external source adapter system is customizable to create from virtually any format the desired keywords. The FOLDOC source is in a textual representation, which holds formatting metadata. The implemented parser for the FOLDOC text is based on several observations, which became rules. For example one of these rules is described as: a line starting with no trailing white spaces, and containing a few words represents a starting of a new keyword in FOLDOC. The associations between the other keywords are represented with special delimiters, for example a keyword is enclosed in parenthesis. These and other rules help the FOLDOC parser construct a true ontology represented in the Topic Map model.

Another approach to create the FOLDOC Topic Map representation would be to discover automatically the important keywords, phrases and associations between them, as in the case of the conference meta and textual data. Implementing this alternative is far beyond our project, but the current framework could stand as a basis for such an extension.

3.1.3 Topic Map Population Phase

The process of Topic Map Population is by far the most challenging and interesting task. It is configured the same way as the typing topics generator, but the used patterns are based on actual topic instances, like the instances of a Paper topic, or Author topic. The generating templates describe a mapping from every structured metadata record to the specified topic instance.

Even techniques based on a semi-structured or structured data face several morphological and semantic problems. The main problem is identifying the entities across several records. For instance the name of a person could be misspelled in a number of ways, or the order of the first and family name is not universal in many languages. Also the use of addressing like Phd., Dr., Msc. can be an obstacle for successful identification. We implemented several language dependent heuristics for tackling misspelling and other problems, but besides this there is also a special pattern file which encapsulates domain and data specific errors. This task uses a multi-phase approach and the heuristics are fired against the data model iteratively. Thus the underlying knowledge representation becomes more and more coherent after every pass.

Besides this first technical part of the populating process, which is based on metadata, there is another task which is based on the raw texts of the papers. These texts contain inherent associations not published explicitly through metadata. For instance the paper is associated to the categories described by keywords. Another example is that one author references another author in the text.

The automated document classification is implemented in a pluggable way. It can use several techniques from the field of unsupervised document classification and statistical information retrieval. To leverage the current implementations of such techniques, we integrated our tools with the WEKA framework.

The simplest approach to assign classifying keywords to papers would be a full text search based classification, for example by searching the abstract of each paper for the keywords of FOLDOC. This approach gives a heavily expanded classification, because every word occurrence is weighted equally. Although this gives rough estimation of used keywords, the final classification results are not acceptable. Using a pipeline architecture we managed to create chains of processing as shown on Figure 2. The original classification created by the simple search based approach is used, and refined in the second part. Using a Vector Space Model (VSM) of the papers, we managed to create a more accurate classification. The vectors in this model were the keywords found by the full text search and every paper could be represented as an element in this space, based on the relative relevancy of every occurring keyword in that particular paper. After conducting several classification experiments, we decided to create an association between the paper and a keyword if and only if the relative relevancy is in the first 66% among the other keywords present in the text of the paper. The magic number of 66% was decided intuitively: full inclusion was too much, half inclusion produced bad results, and the magic number seemed to be acceptable.

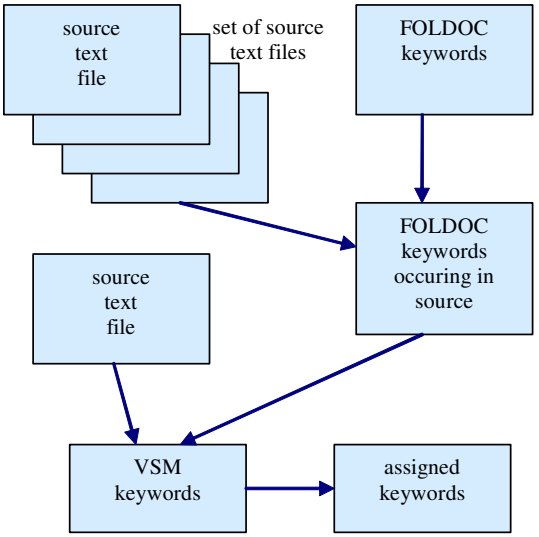


Figure 2. Chains of processing to find the most relevant topics in a paper

This combined technique is language independent, as far as the keywords and the words in the texts are correctly stemmed. Because FOLDOC was currently available in English, we used the English abstract of the papers. By constraining the FOLDOC keyword database to the subset found as a true occurrence in the conference papers, the translation is much easier.

3.1.4 Informational Portal / Content Management Backend

The tangible part, as far as the end user is concerned is the informational web-based portal. The main reason why we have chosen the Topic Map model to provide an abstraction for data representation is that the Topic Map representation of metadata and

the typing topics can easily be published on a portal. The core concepts of this portal are the topics and their associations. The user can navigate from one topic to another as in a Wiki from one page to another. From every type of topic all possible associations are visible and hyperlinkable. From a given author one can access all the authors, the conferences on which the author published and the keywords the author wrote about. By navigating to a selected keyword, all the associated papers are shown.

There are numerous generic frameworks to visualize and to publish topic maps. The Ontopia Omnigator [32] or the open-source TM4Web [35], like any other Topic Map application, are based on a Topic Map engine. The View part consists of easily modifiable Html templates, using a template engine like Jakarta Velocity [31] or standard JSP technology.

While the Topic Map engines fit well as a generic visualization, for the individual portals one must implement the whole navigation and view according to its design and concept. This is why we have chosen another approach to configure the whole portal through a content management backend by leveraging the representational metadata. This metadata is encapsulated in every typing topic and their templates. Content navigation and rendering is also based on typing topics.

In conclusion, the framework implemented on a Topic Map model is content management that leverages knowledge representation. It integrates several general editors that discover the actual type of modifiable data, and chose the appropriate template for the topic being edited or viewed. It uses the built in multi language support of the Topic Map paradigm, which was one of the priorities of this project.

As technology basis, we use the Tapestry web framework [22][23][30], which is a component oriented framework. We have implemented several generic components for every element of the Topic Map paradigm: the Topics, Associations, Occurrences and also lists powered by tolog queries [37] which are topic map queries similar to SQL, but having a Prolog like syntax.

We have extended this generic topic editing and managing framework to refine the specific tasks for specific topic types. Thus we implemented an interface which allows that a topic in administration mode is not only editable, but the user can perform specific tasks. The user is able to rerun the occurrence finder, the classifier or any other implemented action for the current type of topic.

4 CONCLUSION

This project tackled several technological and algorithmic challenges. It investigated the applicability of the Topic Map model on real data. We have experimented with different document classification algorithms and implemented a content management system based on this knowledge representation.

The framework for generating Topic Maps presented in this paper does not assume that the source text has uniform and formally defined structure, it handles English and Hungarian text as well, incorporates machine learning and information retrieval techniques in the process, and provides a flexible content management system for the presentation of the generated Topic Map on a web based portal.

At the time of the writing of this paper an initial Topic Map is generated and tested. The generated Topic Map contains 3537 topics, 723 papers, 973 keywords from ten years of Networkshop conferences. Manual annotation at this scale is not feasible, because the annotation is sometimes regenerated or incrementally extended at each year's conference. There are about eleven thousand keywords in FOLDOC, and in general we do not expect that the number of keywords would dramatically increase. The tools of the framework produce results in seconds when applied to the conference papers of the Networkshop series.

The project has proven the applicability of the Vector Space Model in categorization by reducing with an order of magnitude the irrelevant classifications and keywords. The deployed Topic Map portal is under test. Compared to the original conference web site, the Topic Map portal is user friendly and helps finding the relevant information in ten year's volumes of the Networkshop conference proceedings. The Topic Map generating process is semi automatic which allows the easy incorporation of coming volumes of the conference proceedings.

4.1 Future Work

A future improvement of the classification phase would be the usage of true learning based classifiers, such as Bayes classifiers or others alike. The occurrence or keyword discovery could be made directly from the textual data using advanced keyword and context extraction techniques. A viable solution would be integrating KEA [36], an open-source keyword extraction package, with the current framework.

Another, more visually appealing feature would be an interactive web-based or desktop GUI that guides the end user through the phases. The current Content Management system is generic enough, but it doesn't have yet the necessary abilities to create a full topic map from scratch. At least an ontology must be present in the model. To fully use the potential of Topic Maps, the internal portal metadata could be expressed in terms of Topic Map elements. Thus a generic editor could edit the system itself, if it is carefully configured.

ACKNOWLEDGEMENTS

We are thankful to the Hungarian National Information Infrastructure Development Program for participating in the project, specifying the requirements, giving advices and providing the source of the Networkshop conference series organised by them.

The framework presented in this paper was developed in the Topicportal project supported by the Hungarian Economic Competitiveness Operative Programme (GVOP AKF) under the GVOP-3.1.1-2004-05-0404/3.0 contract.

The project was initiated during the discussions we had with Steve Pepper and finally supported by Ontopia with a special license of the Ontopia Knowledge Suite for this project.

REFERENCES

[1] StevePepper, Lars Marius Garshol - The XML Papers: Lessons on Applying Topic Maps.
http://www.ontopia.net/topicmaps/materials/xmlconf.html

[2] Geir Ove Gronmo - Automagic Topic Maps
http://www.ontopia.net/topicmaps/materials/automagic.html

[3] TMHarvest http://www.folge2.de/topicmaps/tmharvest/userdoc01/en/index.html#features
[4] Lars Marius Garshol - Living with topic maps and RDF http://www.ontopia.net/topicmaps/materials/tmrdf.html
[5] Ora Lassila and Ralph Swick - Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 22 February 1999. Available from http://www.w3.org/TR/REC-rdf-syntax/
[6] ISO/IEC 13250:2000 Topic Maps, International Organization for Standardization, Geneva. Available from http://www.y12.doe.gov/sgml/sc34/document/0129.pdf
[7] Steve Pepper and Graham Moore (editors) - XML Topic Maps (XTM) 1.0, TopicMaps.Org. Available from http://www.topicmaps.org/xtm/1.0/
[8] Graham Moore - RDF and TopicMaps: An Exercise in Convergence, presented at XML Europe 2001 in Berlin. Available from http://www.topicmaps.com/topicmapsrdf.pdf
[9] Lars Marius Garshol - Topic maps, RDF, DAML, OIL. Available from http://www.ontopia.net/topicmaps/materials/tmrdfoildaml.html
[10] Steve Pepper - Ten Theses on Topic Maps and RDF. Available from http://www.ontopia.net/topicmaps/materials/rdf.html
[11] Marc de Graauw 2002 - Survey of Actual Scope Use in Topic Maps. Available from http://www.marcdegraauw.com/files/scope_survey.htm
[12] Marc de Graauw - Structuring Scope. Available from: http://www.marcdegraauw.com/files/structuring_scope.htm
[13] Steve Pepper, Geir Ove Gronmo - Towards a General Theory of Scope. Available from http://www.ontopia.net/topicmaps/materials/scope.htm
[14] Robert Barta, 2003 - Is He The One? Subject Identification in Topic Maps. Available from: http://topicmaps.it.bond.edu.au/docs/21/toc
[15] TMAPI - http://tmapi.org/
[16] Apache Ant - http://ant.apache.org/
[17] Eric Freese - So why aren't Topic Maps ruling the world?, in Extreme Markup Languages 2002: Proceedings. Available: http://www.mulberrytech.com/Extreme/Proceedings/html/2002/Freese01/EML2002Freese01.html
[18] van Rijsbergen, C. J. Information retrieval. Butterworths, 1979.
[19] Salton, Gerard. - Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
[20] Ian H. Witten, Eibe Frank - Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)
[21] Weka - http://www.cs.waikato.ac.nz/ml/
[22] Tapestry - http://jakarta.apache.org/tapestry/
[23] Howard M. Lewis Ship – „Tapestry in Action", Manning Publications Co. (2004), ISBN 1-932394-11-7
[24] FOLDOC http://foldoc.org/
[25] ODP - http://dmoz.org/
[26] Babel - http://www.geocities.com/ikind_babel/babel/babelsr.html
[27] Szószablya - http://mokk.bme.hu/projektek/szoszablya/
[28] YALE - Yet Another Learning Environment http://www-ai.cs.uni-dortmund.de/SOFTWARE/YALE/index.html
[29] jBNC - Bayesian Network Classifier Toolbox http://jbnc.sourceforge.net/
[30] Ka Iok Tong - Enjoying Web Development with Tapestry, ISBN: 1411649133
[31] Jakarta Velocity – http://jakarta.apache.org/velocity/
[32] Ontopia Omnigator – www.ontopia.net/omnigator/
[33] TM4J – http://tm4j.org/
[34] TinyTIM – http://tinytim.sourceforge.net/
[35] TM4WEB – http://tm4j.org/tm4web.html
[36] KEA - http://www.nzdl.org/Kea/
[37] Tolog - http://www.ontopia.net/topicmaps/materials/tolog.html
[38] NIIF Networkshop conference series http://www.iif.hu/rendezvenyek/networkshop/

Graph Retrieval with the Suffix Tree Model

Mathias Lux[1] and **Sven Meyer zu Eissen**[2] and **Michael Granitzer**[3]

Abstract. The paper in hand presents an adoption of the suffix tree model for the retrieval of labeled graphs. The suffix tree model encodes path information of graphs in an efficient way and so reduces the size of the data structures compared to path index based approaches, while offering a better runtime performance than subgraph isomorphism based methods. Within a specific use case we evaluate the correlation of the developed method to human judgement and compare the correlation values to other methods. We show that in our use case, which is the retrieval of digital photos annotated with MPEG-7 using the *MPEG-7 Semantic Description Scheme,* the presented algorithm performs better than other methods.

1 INTRODUCTION

Let $G = \langle V, E \rangle$ be a graph, where V denotes the node set and $E \subseteq V \times V$ denotes the edge set. Given a query graph G_q and a graph set $\mathcal{G}$, graph retrieval deals with the task to identify a subset $\mathcal{R} \subseteq \mathcal{G}$ with the property

$$\forall G \in \mathcal{R} : \varphi(G_q, G) \geq t$$

where $\varphi : \mathcal{G} \times \mathcal{G} \rightarrow \mathbf{R}$ denotes a similarity function and $t \in \mathbf{R}$ is a minimum similarity threshold.

The research question how to search similar graphs in a database was already prescribed in a work by Simmons in 1966 (see [13]), in which he matched conceptual graphs. Since then, different applications areas emerged; they include querying chemical graph databases that store molecular structures, retrieving vector and raster images using characteristics encoded in a graph, and recently, searching in semantically enriched data in the context of semantic Web applications.

Our application scenario relates to multimedia retrieval with the MPEG-7 standard, where metadata are represented as graphs: A user formulates his or her information need in the form of a graph, which is then matched against an MPEG-7 graph database $\mathcal{G}$.

A property of MPEG-7 graphs is that their nodes and edges are labeled with text, say, for each $G \in \mathcal{G}$ there exists a function $l_E : E \rightarrow T_E$ as well as $l_V : V \rightarrow T_V$, where T_E, T_V are term sets. The goal is to retrieve graphs that match both, the query graph's structure as well as the labels. The challenges in this connection are twofold:

1. The statement of a similarity function φ that reflects the application scenario, and
2. The operationalization of the retrieval functionality.

The second challenge restricts the flexibility in formulating a similarity function: φ must not be expensive to evaluate in terms of runtime complexity since in our case a user waits actively for retrieval results.

[1] University of Technology Graz, Knowledge Management Institute, Austria, email: mathias.lux@tugraz.at
[2] Bauhaus University Weimar, Germany, email: sven.meyer-zu-eissen@medien.uni-weimar.de
[3] Know-Center Graz, Austria, email: mgrani@know-center.at

2 RELATED WORK

Although maximum common subgraph isomorphism is a natural starting point for graph similarity computation (see [3]), it cannot be applied to our scenario: First, the question if two graphs G and H contain an isomorphic subgraph whose edge set has more than $k \in \mathbf{N}$ elements is NP-complete (see [6]). Second, quantifying similarity using ratios of subgraph edge set sizes solely may not reflect our problem, since edge label matches can be of different importance, depending on the value of an edge label.

For this and similar reasons, graph retrieval algorithms are tailored to the requirements of the underlying use case. For example, Fonseca et al. used graph invariants of trees—in this specific case the eigenvalues of the tree's and subtree's adjacency matrix—to identify relevant cliparts represented as trees, representing adjacency and inclusion of color areas within the cliparts, in a database (see [5],[12]).

Zong et al. (see [18]) retrieved labeled graphs using an index in which the labels of paths up to a certain length were stored. The relevance between a query graph and a graph from the database was computed from a TF*IDF-like similarity measure that was applied to the edge labels.

Berreti et al. (see [2]) extracted information on neighbouring colour regions from raster images, which was encoded in directed labeled graphs. To retrieve similar images a graph database was queried employing a tailored metric, which proved as slow but highly configurable.

2.1 Contribution

Text retrieval methods based on the vector space model, especially those using inverted lists as described in [1], have been applied to graph retrieval before: A graph's labels form a virtual document; likewise, the query graph's labels are used to construct a query document. The similarity between these documents is computed using the vector space model along with standard similarity measures like TF*IDF or BM-25.

Unlike traditional vector space approaches our proposed method employs the suffix tree model, described in [8]. Its advantage is that similarity computations incorporate word order within sentences and text fragments. Applied to the outlined MPEG-7 retrieval scenario, this property is especially

useful when matching labels in a graph's paths, yielding to better similarity values like the respective experiments show.

3 APPLICATION SCENARIOS

The specification of semantics often follows a graph modeling approach; the pioneering work of Sowa (see [14]) is one of many examples. Similarity search in this and related contexts reduces to graph retrieval.

Currently a trend towards a semantically enriched Web can be noted. This movement started with the vision of a semantic Web by Berners-Lee (see e.g. foreword in [4]) and resulted in the definition of a syntax for semantics, formally defined in an ontology language based on the Resource Description Framework (RDF), which uses a model based on directed labeled graphs.

Another initiative, aimed at an interoperable standards for multimedia data, is the Moving Picture Expert Group, in short MPEG. Within their Multimedia Content Description Interface, short name MPEG-7, they defined a way to semantically describe the contents of multimedia files by interconnecting semantic objects (e.g. agents, places, and so on) by typed semantic relations (see [7] for more details), which again results in directed labeled graphs that encode semantics.

All of the above mentioned scenarios model semantics with directed labeled graphs. While the same edge label can be used more than once within a graph, we assume that node labels are unique within a graph as defined in MPEG-7, RDF and conceptual graphs.

4 APPLYING THE SUFFIX TREE MODEL TO GRAPH RETRIEVAL

Information retrieval methods that have been used in the past for graph retrieval have in common that they transform database graphs $G_i \in \mathcal{G}$ as well as query graphs G_q to documents d_i and d_q, respectively, which are then compared using their vector space model representations in combination with a related similarity measure like the cosine similarity. Here, the documents consist of sentences, which are made up of node and edge label concatenations from paths in the corresponding graphs. This methodology raises two questions:

1. Which paths of a graph should be used for the construction of d_i and d_q?
2. Which retrieval methodology should be chosen for query matching?

With respect to point (1), some heuristics have been proposed. One prominent method is discussed in connection with *GraphGrep* (see [11]). The paths of a graph are extracted either by identifying all paths in a graph up to a certain length, e.g. with a depth first or breadth first search starting from each vertex (see e.g. [15]), or by identifying frequent substructures within the graphs (see e.g. [17] or [16]).

The focus of our research refers to point (2). Known graph retrieval methods that rely on the vector space model disregard term order or include only partial term order information when using n-grams for indexing. In the following, a similarity measure is presented that tackles the aforementioned problem; it compiles *full* path label order information into the similarity values while keeping the computational complexity bounded by a linear function. In this connection, knowledge about suffix trees is necessary prerequisite; some details are summarized in the next section.

4.1 Suffix Trees

The ith suffix of a document $d = w_1 \dots w_m$ is the substring of d that starts with word w_i. A suffix tree of d is a labeled tree that contains each suffix of d along a path whose edges are labeled with the respective words. The construction of a suffix tree is straightforward: The ith suffix of d is inserted by checking whether some edge emanating from the root node is labeled with w_i. If so, this edge is traversed and it is checked whether some edge of the successor node is labeled with w_{i+1}, and so on. If, in some depth k, a node n without a matching edge is reached, a new node is created and linked to node n with an edge labeled with w_{i+k}.

Figure 1 illustrates a the suffix tree in which the documents d_1 ="Boy plays chess" and d_2 ="Boy plays bridge too" have been inserted.

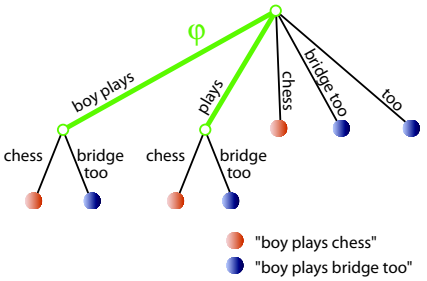


Figure 1. A suffix tree in which the documents d_1 ="Boy plays chess" and d_2 ="Boy plays bridge too" have been inserted.

4.2 Path-based Graph Suffix Trees

Let d_i denote the document that is associated with G_i, and likewise, let d_q denote the document that is associated with G_q. Both, d_i and d_q consist of "sentences", which are concatenations of path labels from selected paths from G_i and G_q, following a heuristic mentioned above.

A natural similarity measure between d_i and d_q arises when inserting each suffix from each sentence of d_i and d_q into an initially empty suffix tree $G_S = \langle V_S, E_S \rangle$. Let $E_i \subseteq E_S$ denote the set of the edges that have been traversed when all suffixes of d_i's sentences have been inserted into G_S, and analogously, let $E_q \subseteq E_S$ denote the traversed edge set for all sentences' suffixes from d_q. The similarity between d_i and d_q can be measured by how many edges E_i and E_q have in common, e.g. quantified by the Jaccard coefficient:

$$\varphi_S(G_i, G_q) = \frac{| E_i \cap E_q |}{| E_i \cup E_q |}$$

Furthermore in [8] two more weighting schemes using term frequency and inverse document frequency of edges, are described to enhance relevance and precision. For similarity calculation of graphs such a weighting can be applied.

In addition to the two original weighting schemes a third scheme relying solely on IDF can be introduced. Stripping the term frequency from the original weighting formula, a similarity measure can be defined as follows:

$$\varphi_{idf}(G_i, G_q) = \frac{1}{\mid E_S \mid} \sum_{e \in E_S} traversed(e) \cdot IDF(e)$$

$$\text{with } traversed(e) = \begin{cases} 0 & e \notin E_i \cap E_q \\ 1 & e \in E_i \cap E_q \end{cases}$$

Here, $IDF : E \rightarrow \mathbf{R}$ is defined to be the inverse document frequency function, $IDF(e) = \log(\frac{n}{S(e)})$, with n being the total number of documents and $S : E \rightarrow \mathbf{N}$ denoting the function that delivers the number of distinct documents that traversed a given edge on insertion into the suffix tree.

5 EVALUATION

Although the presented suffix tree model for graphs can be applied to arbitrary graphs with node and edge labels, the evaluation was done within a multimedia retrieval scenario: Using MPEG-7, the *Multimedia Content Description Interface*, multimedia documents can be annotated using graphs expressing the semantics of the multimedia document. This particular functionality of MPEG-7 is defined in the Semantic Description Scheme (see [7] for details on MPEG-7).

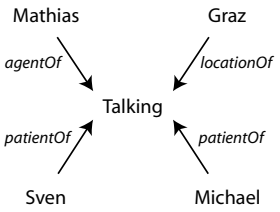


Figure 2. Illustration of an MPEG-7 based annotation expressing that *Mathias is talking to Sven and Michael in Graz.*

Within this scenario two graphs, like the one shown in figure 2, can be compared and a similarity value can be obtained. Based on the used mechanism for similarity calculation different results are achieved. Our evaluation aims to identify the most semantic method (in terms of human judgement) for similarity calculation of MPEG-7 based annotations.

To evaluate the *semantics* of candidate similarity measure a test set of 96 manually annotated digital photos was used. In essence for all photos a labeled directed graph exists, which describes the semantics of the image by specifying persons, time points, locations and events as nodes and interconnecting these nodes by labeled edges, like shown in figure 2. The graphs have a median number of nodes of 5.81, with a medium number of 5.99 edges. From this test data set 20 photo pairs were identified, which were used to create a questionnaire. The participants of the evaluation were asked to rate the pairwise similarity of the photos. The averaged similarity from the participants answers was correlated to the results of the candidate similarity measures.

After initial evaluations of 18 and 15 participants a final evaluation with 112 participants was carried out. The results

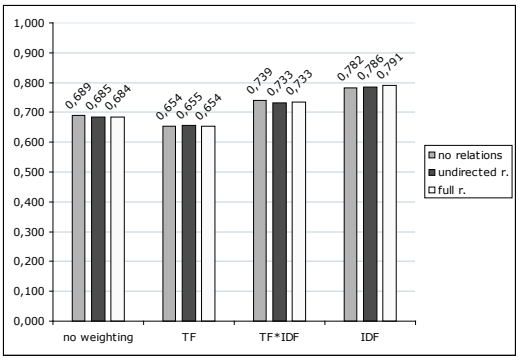


Figure 3. Evaluation of the Suffix Tree Metric in correlation to human judgement

of the evaluation of the suffix tree model based metrics is shown in figure 3. With each weighting scheme three different strategies for building the tree are evaluated: A first approach is to build the tree without taking the edge labels into account (shown as option *no relations* in figure 3), so only the sequence of node labels is inserted into the tree. A second approach is to *normalize* all relation labels without taking their directions into account (shown as option *undirected r.* in figure 3). This can be done by ignoring all direction information on edges. The third option is to use the full paths including node and edge labels (shown as option *full r.* in figure 3).

As can be seen easily the suffix tree model cannot provide an optimal approximation of human judgement with any of the presented weighting schemes. With no weighting schema a rounded maximum correlation value of 0.689 can be achieved. With the term frequency weighting, which was proposed in the original publications the correlation value even gets worse. The inverse document frequency (IDF) weighting proposed in this publication offers the best correlation with a maximum value of 0.791 taking all node and edge information (labels and direction) into account.

Besides the above introduced suffix tree similarity measure for graphs following similarity measures from text and graph retrieval were compared to human judgement:

1. Vector space based on node and edge labels, cosine coefficient as similarity measure with following weighting schemes. This metric does not take the structure of the graph into account, the set of labels is treated as text document:
   (a) without weighting scheme (*Text VS* in fig. 4)
   (b) TF*IDF (*Text VS TF*IDF* in fig. 4)
   (c) BM25 (*Text VS BM25* in fig. 4, see [9] and [10] for details on BM25)
2. Vector space with graph paths as terms, cosine coefficient as similarity measure with following weighting schemes:
   (a) TF*IDF on paths with one arc (*VS IDF Triple* in fig. 4) and full length paths (*VS IDF Paths* in fig. 4)
   (b) BM25 on paths with one arc (*VS BM25 Triple* in fig. 4) and full length paths (*VS BM25 Paths* in fig. 4)
3. Maximum common subgraph metric from [3] (*MCS* in fig.

4)

4. Error correcting subgraph isomorphism metric from [2] with boolean edge label distance functions and two options for used node label distance functions:
   (a) Boolean distance function (*Berretti (Bool)* in fig. 4)
   (b) Term vector distance function (*Berretti (VS)* in fig. 4)

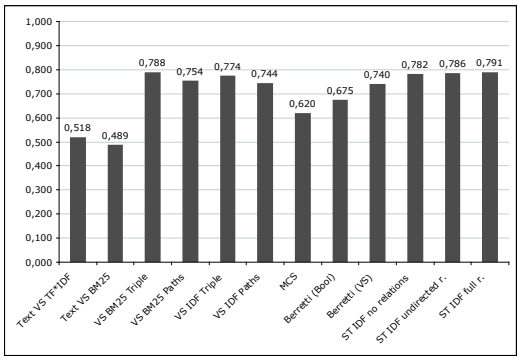


Figure 4. Evaluation of different distance functions and metrics using the correlation to human judgement

The evaluation results in figure 4 show that the suffix tree model with proposed inverse document frequency weighting offers the best correlation to human judgement in the presented domain. However the *VS BM25 Triple* metric offers a nearly as high correlation value. The two variants of the error correcting subgraph isomorphism metric of [2] do not perform as good as the other candidates. All evaluated text based similarity and distance measures, which do not take the structure in to account, do not correlate well with human judgement.

6 CONCLUSION

As can be seen easily from the evaluation similarity measures, which take the structure information of the graphs into account, are superior to the tested text retrieval mechanisms, which use node and edge labels for retrieval. The suffix tree method has a slightly better correlation coefficient and therefore reflects human judgement better than the other methods. However the difference to the vector space method is marginal, which justifies for example the usage of an path index for graph retrieval. One possible explanation why the triple based VS approach performs that good is that in the inspected domain all node labels are unique within a single graph.

The most interesting point is, that methods adapted from text retrieval perform better than the evaluated methods developed for graphs, like MCS and the algorithm of Berretti et al. described in [2] on the used test data set. However the number of photos in the set is too small for general conclusions, but as no test data sets for semantic annotations currently exist, the creation of semantic annotations for multimedia documents is a laborous task and the usefulness of random graphs for evaluation is limited in this domain, an evaluation with a bigger data set was out of scope of the project. Nevertheless the presented evaluation provides a starting point for further investigations.

ACKNOWLEDGEMENTS

The Know-Center is funded by the Austrian Competence Center program K plus under the auspices of the Austrian Ministry of Transport, Innovation and Technology (http://www.ffg.at/index.php?cid=95) and by the State of Styria.

REFERENCES

[1] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc., 1999.

[2] S. Berretti, A. Del Bimbo, and P. Pala, 'A graph edit distance based on node merging', in *Image and Video Retrieval: Third International Conference, CIVR 2004*, volume 3115 of *LNCS*, pp. 464–472, Dublin, Ireland, (July 21-23 2004). Springer.

[3] Horst Bunke and Kim Shearer, 'A graph distance metric based on the maximal common subgraph', *Pattern Recognition Letters*, **19**(3-4), 255–259, (1998).

[4] Dieter Fensel, James A. Hendler, and Henry Lieberman, *Spinning the Semantic Web Bringing the World Wide Web to Its Full Potential*, MIT Press, 2005.

[5] Manuel J. Fonseca, B. Barroso, and Joaquim A. Jorge, 'Retrieving clipart images by content', in *Image and Video Retrieval: Third International Conference, CIVR 2004*, volume 3115 of *LNCS*, pp. 500–507, Dublin, Ireland, (July 21-23 2004). Springer.

[6] Michael R. Garey and David S. Johnson, *Computers and Intractability*, W.H. Freeman and Company, New York, 1979.

[7] Harald Kosch, *Distributed Multimedia Database Technologies*, CRC Press, Nov. 2003.

[8] Sven Meyer zu Eissen, Benno Stein, and Martin Potthast, 'The suffix tree document model revisited', in *Proceedings of the I-Know '05 5th International Conference on Knowledge Management*, pp. 596–603, Graz, Austria, (July 2005). J.UCS.

[9] S. E. Robertson and S. Walker, 'Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval', in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 232–241, New York, NY, USA, (1994). Springer-Verlag New York, Inc.

[10] Stephen Robertson, Hugo Zaragoza, and Michael Taylor, 'Simple bm25 extension to multiple weighted fields', in *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pp. 42–49, New York, NY, USA, (2004). ACM Press.

[11] Dennis Shasha, Jason T. L. Wang, and Rosalba Giugno, 'Algorithmics and applications of tree and graph searching', in *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 39–52. ACM Press, (2002).

[12] Ali Shokoufandeh, Sven J. Dickinson, K. Siddiqi, and S.W. Zucker, 'Indexing using a spectral encoding of topological structure', in *Conference on Computer Vision and Pattern Recognition, IEEE Computer Society*, volume 2, pp. 491–497, USA, (June 1999).

[13] R. F. Simmons, 'Storage and retrieval of aspects of meaning in directed graph structures', *Commun. ACM*, **9**(3), 211–215, (1966).

[14] John F. Sowa, 'Semantics of conceptual graphs', in *Proceedings of the 17th annual meeting on Association for Computational Linguistics*, pp. 39–44, Morristown, NJ, USA, (1979). Association for Computational Linguistics.

[15] Gabriel Valiente, *Algorithms on Trees and Graphs*, Springer, Berlin, Germany, September 2002.

[16] Takashi Washio and Hiroshi Motoda, 'State of the art of graph-based data mining', *SIGKDD Explor. Newsl.*, **5**(1), 59–68, (2003).

[17] Xifeng Yan, Philip S. Yu, and Jiawei Han, 'Graph indexing: a frequent structure-based approach', in *SIGMOD '04: Pro-*

ceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 335–346. ACM Press, (2004).

[18] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu, 'Conceptual graph matching for semantic search', in *ICCS '02: Proceedings of the 10th International Conference on Conceptual Structures*, pp. 92–196, London, UK, (2002). Springer-Verlag.

Common Criteria for Genre Classification: Annotation and Granularity

Marina Santini[1]

ABSTRACT

In this paper, we present two experiments that use machine learning for automatically classifying web pages by genre. These experiments highlight the influence that genre annotation and genre granularity can have on the accuracy of the classification. From a practical point of view these experiments show that a collection annotated with the criteria of 'objective sources' and consistent genre granularity ensures a very good classification accuracy (Experiment 1). Additionally, the classification model built out of such a collection can be exported more profitably for predictive tasks on an unclassified web page collection (Experiment 2). These experiments represent a starting point for a discussion about the need of common criteria for building a genre collection in the absence of an official genre-annotated benchmark.

1 INTRODUCTION

In this paper, we present two experiments that use machine learning for automatically classifying web pages by genre.

Many definitions of genre have been proposed so far in literary studies (e.g. [20]), academic writing (e.g. [23]), professional settings (e.g. [2] and [24]), organizational environment (e.g. [26]), and so on. More specifically, in automatic genre classification studies, genres have often been seen as non-topical categories that could help reduce information overload (e.g. [16] or [15]). In this area, not only text categories such as 'article', 'FAQs', 'home page', etc. have been considered to be genres, but also polarities, such as subjective-objective and positive-negative ([7]), and style ([1], [9] and [5]). Regardless the different definitions and connotations, a classification by genre has been acknowledged to be useful in information retrieval (e.g. [9], [12], etc.), information filtering ([7]), digital libraries ([19]) and other practical applications.

In this paper we present two experiments of genre classification of web pages based on a simplified and intuitive definition of genre, which is suitable for all kind of genres – including genres on the web – and for an automatic approach. In our view, genres can be defined as named *socio-cultural* communication artefacts, linked to a society or a community, bearing standardized traits, leaving space for the creativity of the text producer, and raising expectations in the text receiver. For example, the personal home page (cf. also [6]) has standard traits, such as self-narration, personal interests, contact details, and often pictures related to one's life. However, these conventions do not hinder the creativity of the producer, and as receivers, we expect a blend of standardized information and personal touch. Though unsophisticated, this definition of genre allows us to suggest a practical solution to the main shortcoming in genre classification, i.e. the lack of a genre-annotated benchmark. Because of this lack, the main tendency has always been to build one's own collection according to subjective criteria as for genre annotation and genre granularity. This is especially true for genre studies based on collections of web pages. Although building a genre-annotated benchmark of web pages is difficult and maybe not feasible, because annotating a web page by genre is both hard and controversial (cf. [21]), a few criteria should be discussed and agreed upon. Without some kind of commonality, any comparison becomes unfeasible. For instance, can we state that the 92% accuracy achieved by [3] is better than the accuracy (about 70%) achieved by [17]? The solution we suggest for building more comparable genre collections is to exploit the *socio-cultural* aspect of the concept of genre. As pointed out earlier, genres have a function in a society, culture or community, i.e. they have a social or public role that implies a number of conventions and raises predictable expectations. This means that the role or the function of different genres is recognized and correctly used in the communication interaction. Leveraging on this public and collective acknowledgement it is possible to create a genre-annotated collection without involving human annotators. The key is to download documents from genre-specific archives or portals and use their membership in these containers as an automatic membership in a specific genre. For example, eshops can be randomly downloaded from the portal http://www.eshops.co.uk/ and considered to be eshops without any further manual annotation or inter-rater agreement assessment. We include in the public acknowledgement also genres used as title of documents (for example, "Insects Hotlist"). The idea behind selecting documents with a genre in the title or picking them up randomly from public resources, such as an archives or a portals, is the following: if there is an archive, a portal or a website specialized in, say, pointing to or collecting genres such as eshops, blogs or search engines, this means that the documents pointed to or collected there are considered to belong to these genres by the collectivity of web users. We call this criterion 'annotation by objective sources'. A genre collection annotated by objective sources tends to be more representative as for intra-genre variation than a collection annotated relying on the genre stereotypicality that two, three, or more annotators have in mind. We suggest that annotating a collection using objective sources is faster and closer to real-world conditions.

Genre granularity is also important when building a collection for genre classification. In fact, genre palettes often show different levels of granularity. For instance, [9] includes in his genre palette both FAQs (genre) and journalistic materials (super-genre). We suggest the use of the prototype theory (cf. [18] and [13]) to achieve a consistent level of genre granularity. A prototype is the most typical instance of a more encompassing or fuzzy category. Categories that can be dealt with the prototype theory can be ordered into a three-tiered hierarchy: superordinate level, basic level and subordinate level. For example, the genre 'advertisement' represents the basic level (genre) of the superordinate level 'advertising' (super-genre), while a 'web ad' represents the subordinate level (subgenre) of the basic level. The

[1] University of Brighton (UK); M.Santini@brighton.ac.uk

basic level embodies the information level at which concepts are most easily recognized, remembered and learned with respect to their function. The basic level included in the prototype theory should not be mixed up with document stereotypicality or exemplarity. Building a genre collection choosing exemplars, i.e. only stereotypical documents, to unambiguously represent a genre can return biased results. According to the prototype theory, instead, instances of a genre may vary in their prototypicality, thus allowing intra-genre variation.

The two experiments presented in this paper highlight the influence that genre annotation and genre granularity can have on the accuracy of genre classification of web pages. They were designed to point out several issues (some already covered in [22]). In this paper, these two experiments allow us to emphasize two general aspects of genre classification, one practical and one theoretical. From a practical point of view these experiments show that a collection annotated with the criteria of objective sources and consistent genre granularity ensures a very good classification accuracy (Experiment 1). Additionally, the classification model built out of such a collection can be exported more profitably for predictive tasks on an unclassified web page collection (Experiment 2). From a theoretical point of view, they represent a starting point for a discussion about the need of common criteria in the absence of an official genre-annotated benchmark

In order to ensure replicability, all the materials used for these experiments, including web page collections, feature sets and the manual evaluation of Experiment 2, are available at http://www.nltg.brighton.ac.uk/home/Marina.Santini/, bottom of the page.

The paper is organized as follows: Section 2 provides an overview of recent work in genre classification of web pages; Section 3 presents the web page collections and the two experiments; conclusions are drawn in Section 4.

2 PREVIOUS WORK

Several experiments have been recently carried out with genres and web pages. Here we list the latest studies in order to show how difficult is to compare their results in the absence of common criteria as for corpus building and genre palettes.

[7]: *Number of web pages: 2150; Annotation: single rater; Categories: subjectivity, positive-ness*. They tried to discriminate among texts coming from different domains in terms of two polarities: subjective vs. objective and positive vs. negative. Their aim was to see how a classification model tuned on one domain performed in another domain. According to their results, in single domain classification the best accuracy is achieved with Multi-View-Ensemble (MVE) (see [7] for details) for subjectivity, and with bag-of-words (BOW) features for positive-ness. In domain transfer classification, the best accuracy is achieved with Parts-of-Speech (POS) tags for subjectivity and MVE for positive-ness. Although it is true that genres can be divided into more subjective genres (e.g. editorials), or more objective genres (e.g. surveys), and that the opposition positive-negative can suggest specific genres (such as reviews), these two polarities can hardly be considered as "genres" in themselves. Nonetheless, [7]'s contribution is extremely valuable because they shed some light on the performance of different feature sets across several domains, providing insight into the extent of feature exportability.

[5]: *Number of web pages: 2700; Annotation: one or more raters; Categories: functional styles*. They carried out an experiment on style-dependent document ranking. Their research explored the possibility of incorporating style-dependent ranking into ranking schemata for searching the web and digital libraries. Their basic idea was to reduce styles (more specifically, the five functional styles theorized by the School of Prague) to a single continuous parameter. Regardless the promising preliminary results, they could see little improvement in relevance ranking when stylistic parameters were included.

[3]: *Number of web pages: 343; Genre annotation: the author plus at least one or more raters; Genres: abstract, call for papers, FAQs, hub/sitemap, job description, resume/C.V., statistics, syllabus, technical paper*. She tried out the efficiency of several feature sets and automatic feature selection techniques on a small corpus of 10 genres, using a number of classification algorithms. Although her results can be considered only indicative given the reduced number of pages per genre (an average of 20 web pages per genre class), she made interesting remarks about discrimination across similar genres, and the influence of the genre palette and document exemplarity on discrimination tasks. Her best accuracy (92.1%) was achieved by one of the feature combinations resulting from an automatic feature selection technique.

[10]: *Number of web pages: 321; Genre annotation: do not say; Genres: personal, corporate, organizational home pages, including also non-home pages, as noise*. They tried the hard task of home page genre discrimination. The best accuracy (71.4%) is achieved on personal home pages with a single classifier, manual feature selection, and without noisy pages.

[16]: *Number of web pages: 1224; Genre annotation: two graduate students; Genres: personal home page, public home page, commercial home page, bulletin collection, link collection, image collection, simple table/lists, input pages, journalistic material, research report, official materials, FAQs, discussions, product specification, informal texts (poem, fiction, etc.)*. They investigated the efficiency of several feature sets to discriminate across these 16 genres. They also tested the classification efficiency on different parts of the web page space (title and meta-content, body, and anchors). The best accuracy (75.7%) was achieved with one of their features sets when applied only to the body and anchors.

[17]: *Number of web pages: 800; Genre annotation: three raters; Genres: help, article, discussion, shop, portrayal (non-private), portrayal (private), link collection, download*. They worked out a genre palette of eight genres following the outcome of a study on genre usefulness. As they aimed at a classification performed on the fly, they assessed features according to the computational effort they required, giving preference to those requiring low or medium effort. They achieved around 70% accuracy with discriminant analysis on the palette of eight genres. Other results relate to groups of genres tailored for web user profiles.

[14] and the follow up [15]: *Number of web pages: 321; Genre annotation: at least two raters; Genres: reportage-editorial, research article, review, home page, Q&A, specification*. They aimed at selecting genre-revealing terms from the training document set using collection of web pages annotated both at topic level and at genre level. Their formula (the deviation formula) makes use of both genre-classified documents and subject-classified documents and eliminate terms that are more subject-related than genre-related. They report a micro-average of precision and recall of about 90%.

As already stressed, the absence of common criteria or evaluation ground makes most of these experiments (see Table 1 for a summary) difficult to compare, however fruitful each study can be in itself. A cross-evaluation of these experiments remains virtually unfeasible because genre palettes are mostly disparate. Also in the case of 'home page', which is probably one of the few genres in common in several experiments, any comparison appear to be difficult, because selection criteria and level of exemplarity are not declared. The two criteria of annotation by objective sources and consistent level of granularity are suggested to overcome this un-comparability.

Table 1. Summary Table

Studies	No. of web pages	Annotation	Labels
[7]	2,150	single rater	Subjectivity vs. objectivity, positive vs. negative
[5]	2,700	One or more raters	public affairs style, everyday communication style, scientific style, journalistic style, literary style
[3]	343	Two or more raters	abstract, call for papers, FAQs, hub/sitemap, job description, resume/C.V., statistics, syllabus, technical paper
[10]	321	do not say	home pages (personal, corporate, organizational)
[16]	1,224	two graduate students	personal home page, public home page, commercial home page, bulletin collection, link collection, image collection, simple table/lists, input pages, journalistic material, research report, official materials, FAQs, discussions, product specification, informal texts
[17]	800	3 raters	article, discussion, shop, portrayal (non-private), portrayal (private), link collection, download
[14] and [15]	321	at least two raters	reportage-editorial, research article, review, home page, Q&A, specification

3 EXPERIMENTS

3.1 7-Web-Genre Collection

The *7-web-genre collection* includes 200 English web pages per genre, amounting to a total of 1,400 web pages (available online at the URL reported in the Introduction). These web pages were collected by the author of this paper in early spring 2005. This collection was built with genres belonging to a consistent level of granularity and applying the annotation by objective source. The seven web genres included in the collection are the following:

1. blog
2. eshop
3. FAQs
4. online newspaper front page
5. list
6. personal home page[2]
7. search page

[2] 'Personal home page' is the basic level of the superordinate level 'home page' and has 'academic personal home page', 'administrative personal home page', etc. as subordinate level.

The web pages included in the 7-web-genre collection were randomly downloaded from the following public archives or portals (download date: Feb-March 2005):

- Blogs:
  - http://www.britblog.com/
  - http://www.nataliedarbeloff.com/augustinearchive.html.
- Eshops:
  - http://www.shops.co.uk/
  - http://www.eshops.co.uk/
- FAQs:
  - http://www.cybernothing.org/faqs/net-abuse-faq.html
  - http://www.irs.gov/faqs/
  - http://www.copyright.gov/help/faq/
  - http://www.aoml.noaa.gov/hrd/tcfaq/tcfaqHED.html
- Newspaper front pages belong to a number of different online newspaper and are available at Internet Archive:
  - www.archive.org
- Personal home pages are heterogeneous, and include academic and administrative personal home pages, as well as more informal personal home pages. They were downloaded from:
  - http://dmoz.org/Society/People/Personal_Homepages/
  - http://www.math.unl.edu/~mbritten/ldt/homepage.html
  - http://www.bradley.edu/people/fac-staff.html
  - http://www.daimi.au.dk/local/map/PeopleandLocationsPeopleFrame.html
  - http://www.mit.edu/Home-byUser.html
  - http://dir.yahoo.com/Society_and_Culture/People/Personal_Home_Pages
  - http://hpsearch.uni-trier.de/hp/a-tree/
  - Search pages comes from:
  - http://www.searchenginecolossus.com/

The web pages included in the genre 'list', were selected searching keywords in Google and selecting relevant web pages from the results. All the lists include one of the following keywords (and orthographic variants) in the heading: *checklist, hot list, table of content,* and *sitemap* (see, for example, Insect Hotlist at http://www.fi.edu/tfi/hotlists/insects.html).

3.2 KI-04 corpus

KI-04 corpus was built following a palette of eight genres suggested by a user study on genre usefulness ([17]). It includes 1,295 English web pages (HTML documents), but only 800 web pages (100 per genre) were used in the experiment described in [17]. In Experiment 1, we used 1,205 web pages because some web pages were empty (both original version, 1,295 web pages, and working version, 1,205 web pages, are available online at the URL reported in the Introduction). KI-04 corpus includes:

1. article (127 web pages)
2. download (151 w. p)
3. link collection (205 w. p)
4. portrayal (priv.) (126 w. p)
5. discussion (127 w. p)
6. help (139 w. p)
7. portrayal (non-priv) (163 w. p.)
8. shop (167 w. p)

The KI-04 corpus was collected using bookmarks from about five people. Some genres were extended to get a better balance. The corpus was sorted by three people, one of them wrote a bachelor thesis (in German) on the corpus building process. One of the author of [17] checked many of the pages, and most of the sorting complied with his understanding of the genre categories. The download date was January 26th, 2004.

3.3 SPIRIT collection

The *SPIRIT collection* is a random crawl carried out in 2001 (see [8]). It contains single web pages and not full websites. The size of the whole collection is about one terabyte, and the number of web pages (mostly HTML files) is about 95 millions. It is multilingual and without any meta-information, apart from a short header including the original URL, the date and time when the pages were crawled from the web, and few other details. It represents a genuine slice of the real web. In Experiment 2, we used only **1,000** English web pages (available online at the URL reported in the Introduction) from this random, multilingual and unclassified collection.

3.4 Experiment 1

The practical aim of Experiment 1 was to build two single-label discrete classification models, one out of the 7-web-genre collection, the other from KI-04 corpus, and compare their accuracy results. Both collections were submitted to the same pre-processing. The unit of analysis was a single static web page in HTML format.

The feature set, called *1_set*, used in Experiment 1 includes:

- the 50 most common words in English;
- 24 Part-of-Speech (POS) tags;
- 8 punctuation marks: full stop (.), colon (:), semi-colon (;), comma (,), exclamation mark (!), question mark (?), apostrophe ('), and quotes (");
- genre-specific words[3];
- 28 HTML tags;
- 1 nominal attribute representing the length of the web page (SHORT, MEDIUM and LONG).

(This feature set, together with a description, is available online at the URL reported in the Introduction). The classification algorithm used both in Experiments 1 and 2 is SMO (which implements the Sequential Minimal Optimisation (SMO) for training support vectors) with default parameters and logistic regression model, from Weka machine learning workbench ([25]). Accuracy results, shown in Table 2, are averaged over stratified 10-fold crossvalidations repeated 10 times.

Table 2. Averaged Accuracies with SMO

Averaged Accuracy on the 7-web-genre collection	Averaged Accuracy on KI-04 corpus
90.6%	**68.9%**

As you can see in Table 2, the accuracy of the model built with the 7-web-genre collection is much higher than the model built with KI-04 corpus, namely +21.7%.

In order to see whether the feature set was too tailored or biased towards the 7-web-genre collection, we compared the accuracy of this feature set on KI-04 corpus with the accuracy rates reported in [17]. To make this comparison possible, we ran discriminant analysis using our feature set on KI-04 corpus. As [17] ran their discriminant analysis only on 800 web pages while we used 1,205 web pages, we converted all the results into percentages. A breakdown of the different accuracy rates achieved with discriminant analysis and two different feature set is shown in Table 3.

Table 3. Accuracy rates with discriminant analysis

KI-04 corpus	Our feature set	[17]'s feature set
Article	80.3%	81.3%
Discussion	76.4%	68.5%
Download	74.2%	79.6%
Help	59.7%	55.1%
Link Collection	69.3%	67.6%
Portrayal (non-priv)	59.5%	57.9%
Portrayal (priv)	73.8%	67.7%
Shop	68.3%	66.9%
Accuracy	**70.2%**	**68.1%**

Our feature set performs better than [17]'s feature set. Although the difference is rather small (+2.1%), it is statistically significant (chi-square test). This means that our feature set is not biased toward the 7-web-genre collection, but it performs significantly better than [17]'s feature set on KI-04 corpus with discriminant analysis, i.e. the same algorithm used in [17].

3.4.1 Discussion

Experiment 1 compares the accuracies of two models built with the same classification algorithm, the same feature set but different web page collections, the 7-web-genre collection and KI-04 corpus. The accuracy on the 7-web-genre collection (1,400 web pages) is above 90% while the accuracy on KI-04 corpus is definitely lower. A first thought was that our feature set did not represent the genre palette of KI-04 corpus adequately. However, after having compared the performance of our feature set with [17]'s feature set using the same algorithm (discriminant analysis) on the same collection, we saw that the accuracy achieved by our feature set was slightly higher than the accuracy stated in [17]. Although KI-04 corpus contains eight genres, i.e. one genre more than the 7-web-genre collection (error rate usually increases with the number of categories), this does not justify such a wide the gap in the classification accuracy. Also, it is important to stress that genre-specific words are tailored to the genre palette. This means, the genre-specific words used for the 7-web-genre collection account for blogs, search, front page, etc., while those employed for KI-04 corpus include words relate to articles, discussion, download, etc. Since these two genre palettes have two web genres in common, i.e. home page/portrayal (priv) and eshop/shop, in these two cases the same set of genre-specific words was used for both web genre collections. That the feature set used in the KI-04 corpus is not biased towards the 7-web genre collection is confirmed by the results shown in Table 3, where the performance of our features set is higher than [17]'s feature set.

In conclusion, if neither the feature set nor the classification algorithm is the cause of this large discrepancy in accuracy, then the suspicion is that the selection of the web pages representing genres in KI-04 corpus might be responsible for the lower performance. Although the issue of subjectivity of the assignment of genre to web pages needs further investigation (cf. also [4]), for the time being we interpret the higher performance on the 7-web-genre collection as a result of the application of the two criteria of

[3] Genre-specific words were selected through a cursory manual analysis. A total of 13 sets of genre-specific words were built. 13 and not 15 because two sets were shared across the two collections, namely those related to home-page/portrayal (priv) and eshop/shop. It is worth saying that genre-specific words (available online at the URL reported in the Introduction) are not numerous. For example, genre-specific words for the search web genre are only: *search, crawl, directories, engine, find,* and *see*.

annotation by objective sources annotation and consistent genre granularity.

3.5 Experiment 2

The goal of Experiment 2 was to see whether the classification model built with the collection complying to the criteria of annotation by objective source and consistent genre granularity is more effective also for predictive tasks. In other words, predictions are used here as a kind of evaluation metrics of the efficiency of classification models.

In this experiment we used the two classification models built in the previous experiment together with additional models. The practical aim was to make predictions on unclassified and non-annotated web pages, i.e. 1,000 random English web pages from the SPIRIT collection. The relevance of the agreed upon web pages (see Tables 5 and 6) to a genre was manually assessed by the author of this paper (the breakdown of this manual evaluation is available online at the URL reported in the Introduction).

When making a prediction, the classifier returns a probability score to be interpreted in terms of classification confidence. This confidence score can be exploited when assessing the value of a prediction and for setting a threshold for reliable guesses. In order to get predictions on genre labels which were as reliable as possible, we devised an approach inspired by co-training. The basic idea was to exploit three different views (i.e. three different feature sets) on the same data. When the three models built with the three feature sets agreed on the same genre label (3-out-of-3 agreement) at very high confidence score, namely >=0.9, this was for us an indication of a good prediction. Additionally, as we have two web page collections with two different genre palettes, we can have multi-label predictions. Ideally, a web page might get a prediction of "personal home page", following the palette adopted in the 7-web-genre collection, and "portrayal (private)", following the genre palette adopted in KI-04 corpus. Also, as the two palettes are mostly not overlapping, it is interesting to see which palette is more suitable for the classification of this SPIRIT random sample. From the previous experiment we had two models built with a single feature set (*1_set*). To these models, we add four additional models (two per collection) in order to get the three simultaneous views on each collection. The additional two models were built using the feature sets called *2_set* and *3_se*t (these feature sets, together with a description, are available online at the URL reported in the Introduction).

2_set contains the following features:

- POS trigrams;
- 8 punctuation symbols (as above);
- genre-specific words (as above);
- 28 HTML tags (as above);
- 1 nominal attribute representing the length of the web page (as above).

3_set contains the following features:

- 86 linguistic facets[4];
- genre-specific words;
- 6 HTML facets;
- 1 nominal attribute representing the length of the web page (as above).

[4] Linguistic facets and HTML facets are groups of features highlighting an aspect in the communicative context that is reflected in the use of language. They are listed in the URL reported in the Introduction.

Table 4 shows the performance of the three feature sets on the two web genre collections.

Table 4. Accuracies of three feature sets on two collections

Classification algorithm: Weka SMO	Averaged accuracy on the 7-web-genre collection	Averaged accuracy on KI-04 corpus
1_set	90.6%	68.9%
2_set	89.4%	64.1%
3_set	88.8%	65.9%

From the summary shown in Table 5, we can see that a very low number of pages were agreed upon by the three classification models (second column) built on the 7-web-page collection. This is not necessarily bad when aiming at high precision (future work will explore the possibility of increasing precision).

Table 5. Correct predictions with the 7-web-genre palette

7 WEB GENRE PALETTE	# OF AGREED UPON WEB PAGES (OUT OF 1,000)	CORRECT GUESSES	INCORRECT GUESSES AND UNCERTAIN	ERROR RATE
BLOG	17	1	16	0.94
ESHOP	11	3	8	0.73
FAQs	8	1	7	0.88
FRONTPAGE	7	0	7	1.00
LISTING	18	7	11	0.61
PHP	44	10	34	0.77
SPAGE	12	6	6	0.50
TOTAL	117	28	89	
PERCENTAGE	**11.7%**	**2.8%**	**8.9%**	

However, predictions are even sparer with the models built using KI-04 corpus (Table 6). As there was no 3-out-of-3 agreement for discussion, download, help, and portrayal (non-private), these genres were evaluated with 2-out-of-3 agreement. No correct guesses were returned for article, discussion, download, and help.

Table 6. Correct predictions with KI-04 corpus

KI-04 CORPUS	# OF AGREED UPON WEB PAGES (OUT OF 1,000)	CORRECT GUESSES	INCORRECT GUESSES AND UNCERTAIN	ERROR RATE
ARTICLE	4	0	4	1.00
DISCUSSION	8	0	8	1.00
DOWNLOAD	4	0	4	1.00
HELP	3	0	3	1.00
LINK	3	3	0	0.00
PORTRAYAL (NON-PRIVATE)	5	1	4	0.80
PORTRAYAL (PRIVATE)	7	3	4	0.57
SHOP	6	3	3	0.50
TOTAL	36	10	26	
PERCENTAGE	**3.6%**	**1%**	**2.6%**	

3.5.1 Discussion

Experiment 2 shows that the classification models built with the 7-web-genre collection return a higher number of predictions. This seems to confirm the interpretation that using the two criteria of objective source annotation and consistent level of granularity ensures better classification models and consequently a higher number of correct predictions. Also, this experiment shows a useful methodology to follow for multi-genre classification of web pages, which can be refined and further investigated in future.

4 CONCLUSIONS

In this paper we pointed out how classification models learned from a web collection annotated by genre using the two criteria of annotation by objective source and consistent level of granularity can return higher accuracy and a higher number of correct predictions.

The annotation by objective source is not only less subjective and closer to real-world conditions, but also much faster than annotation by human raters, which is usually time-consuming, controversial, and expensive. Further, a collection built with a consistent level of genre granularity seems to be learned more profitably by the classifier. Together, these two criteria enhance the performance of classification algorithms.

However, a full comparison between the results achieved with the two web page collections built with different criteria is not entirely feasible because the two genre palettes are mostly different. Nonetheless, these findings are indicative of a tendency that can be further investigated in future. It is also worth pointing out that objective sources may still contain biases. Biases in web collections relate to the well-known issue of 'corpus representativeness', dating back to Chomsky's aversion to the use of corpora. However, in the present days and with the web available, biases can be alleviated by randomly picking up web pages from several genre-specific web archives or portals.

Although the two criteria of annotation by objective source and consistent level of granularity represent a practical solution that can help genre classification, the concept of genre remains hard to capture computationally and statistically in its entirety.

First, it would be interesting to investigate more about the ideal proportion among corpus size, number of features and number of classes and its influence on classification results. Also, up to now only single-label discrete classification has been tried out in genre classification studies. Experiment 2 implicitly shows an easy method that can be exploited for multi-label classification: the use of concurrent genre palettes over the same unclassified collection. Ideally, the use of several classification models built with different collections annotated by external sources and a consistent granularity, and including different genre palettes can suggest several genre labels for the same web page. Multi-genre documents and genre hybridism are particularly acute when dealing with web pages, which appear much more unpredictable and individualized than paper documents. Using concurrent genre palettes might represent an alternative to the multi-faceted approach by [11]. What is less reassuring is the absence of a proper evaluation metrics for multi-label problems. We leave these problems open to further investigations and invite the genre classification community to make use of the three collections employed in these experiments and now available online.

5 REFERENCES

[1] Argamon, S., Koppel, M., Avneri, G. Routing documents according to style, *Proc. First International Workshop on Innovative Internet Information Systems*, 1998.

[2] Bathia, V. *Analysing Genre. Language Use in Professional Settings*, Longman, London and New York, 1993.

[3] Boese, E. *Stereotyping the Web: Genre Classification of Web Documents*, M.S. Thesis, Colorado State Univ., 2005.

[4] Boese, E and Howe A. Effects of Web Document Evolution on Genre Classification, *CIKM'05*, 2005.

[5] Bravslavski, P. and Tselischev, A. Experiment on Style-Dependent Document Ranking, *Proc. of the 7th Russian Conference on Digital Libraries*, 2005.

[6] Dillon, A. and Gushrowski, B. Genres and the Web: is the personal home page the first uniquely digital genre?, *JASIS*, 51(2), 2000.

[7] Finn, A. and Kushmerick, N. Learning to classify documents according to genre. *JASIST*, Special Issue, 7(5), 2006.

[8] Joho, H. and Sanderson, M. The SPIRIT collection: an overview of a large web collection, *SIGIR Forum*, 38(2) 2004.

[9] Karlgren, J. *Stylistic Experiments for Information Retrieval*, Thesis submitted for the degree of Doctor of Philosophy, Stockholm University, Sweden, 2000.

[10] Kennedy, A. and Shepherd, M. Automatic Identification of Home Pages on the Web, *Proc. 38 HICSS*, 2005.

[11] Kessler, B., Numberg, G. and Shütze, H. Automatic Detection of Text Genre, *Proc. 35 Annual Meeting of the ACL and 8th Conference of the EACL*, 1997.

[12] Kwasnik, B., Crowston, K., Nilan, M. and Roussinov, D. Identifying document genre to improve web search effectiveness. *The Bulletin of the American Society for Information Science and Technology*, 27(2), 23–26, 2000.

[13] Lee, D. Genres, Registers, Text types, Domains, and Styles: Clarifying the concepts and navigating a path through the BNC Jungle, *Language Learning and Technology*, 5(3), 37-72, 2001.

[14] Lee, Y. and Myaeng, S. Automatic Identification of Text Genres and Their Roles in Subject-Based Categorization, *Proc. 37 HICSS*, 2004.

[15] Lee, Y. and Myaeng, S. Text Genre Classification with Genre-Revealing and Subject-Revealing Features, *Proc. 25 Annual International ACM SIGIR*, 145-150, 2002.

[16] Lim, C., Lee, K. and Kim G., Automatic Genre Detection of Web Documents, in Su K., Tsujii J., Lee J., Kwong O. Y. (eds.) *Natural Language Processing*, Springer, Berlin, 2005.

[17] Meyer zu Eissen S. and Stein B. Genre Classification of Web Pages: User Study and Feasibility Analysis, in Biundo S., Fruhwirth T., Palm G. (eds.), *Advances in Artificial Intelligence*, Springer, Berlin, 256-269, 2004.

[18] Paltridge, B. Working with genre: A pragmatic perspective, *Journal of Pragmatics*, 24, 393-406, 1995.

[19] Rauber, A. and Müller-Kögler, A. Integrating Automatic Genre Analysis into Digital Libraries, *ACM/IEEE joint Conference on Digital Libraries*, Roanoke, USA, 2001.

[20] Rosmarin, A. *The Power of Genre*, University of Minnesota Press, Minneapolis, 1985.

[21] Santini, M. Genres In Formation? An Exploratory Study of Web Pages using Cluster Analysis, *Proc. CLUK 05*, 2005.

[22] Santini, M. Some Issues in Automatic Genre Classification of Web Pages. *Proc. of the JADT 2006* Besançon 2006.

[23] Swales, J. *Genre Analysis*, Cambridge University Press, Cambridge, 1990.

[24] Trosborg, A. (ed.), *Analysing Professional Genres*, J. Benjamins Publishing Company, Amsterdam, 2000.

[25] Witten, I. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, Amsterdam, second edition, 2005.

[26] Yates, J., and Orlikowski, W. Genres of organizational communication: A structural approach to studying communications and media, *Academy of Management Review*, 17(2), 229-326, 1992.

Ensemble-based Author Identification Using Character N-grams

Efstathios Stamatatos[1]

Abstract. This paper deals with the problem of identifying the most likely author of a text. Several thousands of character *n*-grams, rather than lexical or syntactic information, are used to represent the style of a text. Thus, the author identification task can be viewed as a single-label multiclass classification problem of high dimensional feature space and sparse data. In order to cope with such properties, we propose a suitable learning ensemble based on feature set subspacing. Performance results on two well-tested benchmark text corpora for author identification show that this classification scheme is quite effective, significantly improving the best reported results so far. Additionally, this approach is proved to be quite stable in comparison with support vector machines when using limited number of training texts, a condition usually met in this kind of problem.

1 INTRODUCTION

Author identification is the task of predicting the most likely author of a text given a predefined set of candidate authors. This task can be seen as a single-label multi-class text categorization problem [17] where the candidate authors play the role of the classes. Early attempts to author identification focused mainly on cases of disputed authorship [13] or literary works [3] with limited number of candidate authors, sometimes providing controversial results. However, a growing number of studies indicate that the field is now mature to handle difficult cases with many candidate authors and limited number of short training texts [1, 4, 5, 8, 9, 10, 15, 18, 20].

One major subtask of the author identification problem is the extraction of the most appropriate features for representing the style of an author, the so-called *stylometry*. Several measures have been proposed, including attempts to quantify vocabulary richness, function word frequencies and part-of-speech frequencies. A good review of stylometric techniques is given by Holmes [6].

Obviously, the most straightforward approach to represent a text is by using word frequencies, a method widely applied to topic-related text categorization as well. To this end, the most appropriate words for author identification may be selected arbitrarily [13], according to their discriminatory potential on a given set of candidate authors. Burrows [3] first indicated that the most frequent words of the texts (like 'and', 'to', etc.) have the highest discriminative power for stylistic purposes. Interestingly, these words are usually excluded from topic-related text categorization systems. Additionally, this approach for selecting appropriate words is language-independent.

[1] Dept. of Information and Communication Systems Eng., University of the Aegean, 83200 – Karlovassi, Greece, email: stamatatos@aegean.gr

A recent study [9] shows that sub-word units like character *n*-grams (i.e., character sequences of length *n*) can be very effective for capturing the nuances of an author's style. The most frequent *n*-grams of a text provide crucial information about the author's stylistic choices on the lexical, syntactical, and structural level. For example, the most frequent 3-grams of an English corpus indicate lexical ('the', ' to', 'tha', 'con'), syntactical ('ing', 'ed '), or structural ('. T', ' "T') information.

In this paper, we follow the language-independent stylometric approach proposed by Burrows [3] using character *n*-gram frequencies instead of word frequencies. Several thousands of the most frequent *n*-grams are used to repressent the style of a text. From a machine learning point of view, the task of author identification can, then, be viewed as a classification problem of high dimensional feature space (several thousands of valuable features). As proved by previous studies, every word (and subsequently *n*-gram) is valuable for text classification [7]. Therefore, feature selection methods that attempt to reduce the feature set seem not suitable for this task. Moreover, the longer the feature set, the more sparse the data (i.e., the less frequent an *n*-gram, the less likely to be found in a given text).

A machine learning approach able to cope with such a classification task is an ensemble of classifiers based on *feature set subspacing* [2]. That is, to avoid the curse of dimensionality problem, the feature set is divided into smaller parts, each used to train a base learner. The predictions of the base classifiers are, then, combined to provide the most likely class. In this paper, we propose a suitable ensemble-based model and apply it to character *n*-gram representations of authors' style. Comparative performance results are provided for the ensemble-based approach and an alternative model using support vector machines, based on two benchmark text corpora previously used by author identification studies. Moreover, we focus on practical considerations of the task in question, such as limited number of training texts, a condition usually met in real-world author identification problems.

The rest of this paper is organized as follows. Section 2 presents the learning ensemble classification scheme as used in this study. The *n*-gram data sets and the other methods used for comparative purposes are described in section 3. The performance results of the examined schemes are included in section 4. Finally, section 5 summarizes the conclusions drawn and suggests future work directions.

2 CLASSIFICATION SCHEME

In the current approach, each text is represented as a vector of character *n*-gram frequencies of occurrence. Let $\mathbf{G}_d=\{g_1, g_2, \ldots, g_d\}$ be the ordered set (by decreasing frequency of occurrence) of the most frequent *n*-grams (i.e., character sequences of length *n*) of the

training set. Consider f_{ij} as the normalized frequency of occurrence of the *j*-th *n*-gram of $\mathbf{G}_d$ in the *i*-th text. Then, a text x_i is represented as the ordered vector $<f_{i1}, f_{i2}, \ldots, f_{id}>$.

For constructing a classifier ensemble based on feature set subspacing we follow an approach we call *exhaustive disjoint subspacing*. That is, a large feature set is divided into equally-sized disjoint feature subsets drawn at random. Each particular attribute is used exactly once. Each resulting feature subset is used to train a base classifier using a learning algorithm able to provide posterior probabilities. In this study, *linear discriminant analysis* is used. This standard technique from multivariate statistics is a well-known stable classification algorithm proven to be a good compromise between classification accuracy and training time cost [12]. The predictions of the base classifiers are, then, combined based on an appropriate combination method as described in the following subsections.

2.1 Base Classifiers

Let $G_{m:d}$ be a subset of m features drawn (without replacement) at random from the set $\mathbf{G}_d$ of the most frequent *n*-grams of the training corpus ($m \leq d$). Consider $C(G_{m:d})$ as a single linear discriminant classifier trained on the frequencies of these *m* *n*-grams in the training set texts. Then, $E(C(G_{m:d}), combination)$ is an ensemble of such base classifiers according to the *combination* method. When every feature is used exactly once in the framework of an ensemble, we have an exhaustive disjoint subspacing ensemble. In this case, the number of base classifiers is *d/m*. Preliminary experiments indicated that the lower the *m*, the better (and more stable) the performance of the ensemble model. In the experiments described in this study, feature subsets of minimal length are used (m=2).

Consider **L** as the set of all possible classes (authors), then the *i*-th classifier assigns a posterior probability $P_i(C_i(G_{m:d}), x, c)$ to an input text *x* for each $c \in \mathbf{L}$, so that

$$\sum_{j=1}^{|L|} P_i(C_i(G_{m:d}), x, c_j) = 1$$

where $|\mathbf{L}|$ is the size of **L**. In case of learning algorithms that provide crisp predictions, the posterior probabilities can only take binary values (0 or 1).

2.2 Combination Method

Provided the posterior probabilities of the constituent classifiers, an ensemble assigns a posterior probability to an input text for each class according to the combination of the predictions of the base classifiers. Commonly, a combined decision is obtained by just averaging the estimated posterior probabilities (the *mean* rule):

$$P(E(C(G_{m:d}), mean), x, c) = \frac{1}{k}\sum_{i=1}^{k} P_i(C_i(G_{m:d}), x, c)$$

where *k* is the number of the base classifiers. Recall that for exhaustive disjoint subspacing $k=d/m$. Given that the base classifiers are based on different feature sets, their decisions are considered to be independent. When the Bayes theorem is adopted, an alternative combination rule can, then, be applied to the outputs of the base classifiers (geometric mean or the *product* rule):

$$P(E(C(G_{m:d}), product), x, c) = \sqrt[k]{\prod_{i=1}^{k} P_i(C_i(G_{m:d}), x, c)}$$

Comparison of these two combination rules has shown that under the assumption of independence the product rule should be used. However, in case of poor posterior probability estimates, the mean rule is proved to be more fault tolerant [19].

In this study, we use a combination of these two combination rules (henceforth called *mp*). The *mp* rule is just the average of *mean* and *product* rules. Note that the *mean* rule is affected by high values of posterior probabilities, therefore it is favorable for cases where a few base classifiers have assigned a high posterior probability to a class. On the other hand, the *product* rule is affected by low values of posterior probabilities, therefore it is favorable for cases where only a few base classifiers have assigned low posterior probability to a class. Hence, *mp* is a good compromise of these two.

To complete the classification model, provided that $label(classifier, instance)$ is the class assigned by a classifier to a test instance, then, a classifier ensemble chooses the class that maximizes the posterior probability for an input text *x*, that is:

$$label(ensemble, x) = \arg\max_{c \in L}(P(ensemble, x, c))$$

2.3 Effectiveness Measures

The performance of a classifier ensemble is directly measured by the classification accuracy on the test set. Moreover, the effectiveness of an ensemble is indirectly indicated by the diversity among the predictions of the base classifiers as well as the accuracy of the individual base classifiers. In particular, many measures have been proposed to represent the diversity of an ensemble [11]. In this study, the *entropy* measure is used, that is:

$$entropy = \frac{1}{|T|}\sum_{i=1}^{|T|}\sum_{c=1}^{|L|} -\frac{N_c^i}{k}\log_{|L|}\left(\frac{N_c^i}{k}\right)$$

where *k* is the number of base classifiers, $|\mathbf{T}|$ is the total number of test texts and N_{ic} is the number of base classifiers that assign text *i* to class *c*. Notice that log is taken in base $|\mathbf{L}|$ to keep the entropy within the range [0,1]. The higher the entropy of an ensemble, the more diverse the predictions of the individual constituent classifiers.

3 EXPERIMENTAL SETTINGS

3.1 Data Sets

The text corpora used in this study are two well-tested benchmarks for authorship identification. In particular, the texts were published within 1998 in the Modern Greek weekly newspaper *TO BHMA* (the tribune), and were downloaded from the WWW site of the newspaper. The texts are divided into two groups of authors:

- **Group A** (hereafter GA): It consists of ten randomly selected authors whose writings are frequently found in the section A of the newspaper. This section comprises texts written mainly by journalists on a variety of current affairs. Moreover, for a certain author there may be texts from different text genres (e.g., editorial, reportage, etc.). Note that in many cases such texts are highly edited in order to conform to a predefined style, thus washing out specific characteristics of the authors which complicate the task of attributing authorship.
- **Group B** (hereafter GB): It consists of ten randomly selected authors whose writings are frequently found in the section B of the newspaper. This supplement comprises essays on science, culture, history, etc. in other words, texts in which the idiosyncratic style of the author is not overshadowed by functional objectives. In general, the texts included in the

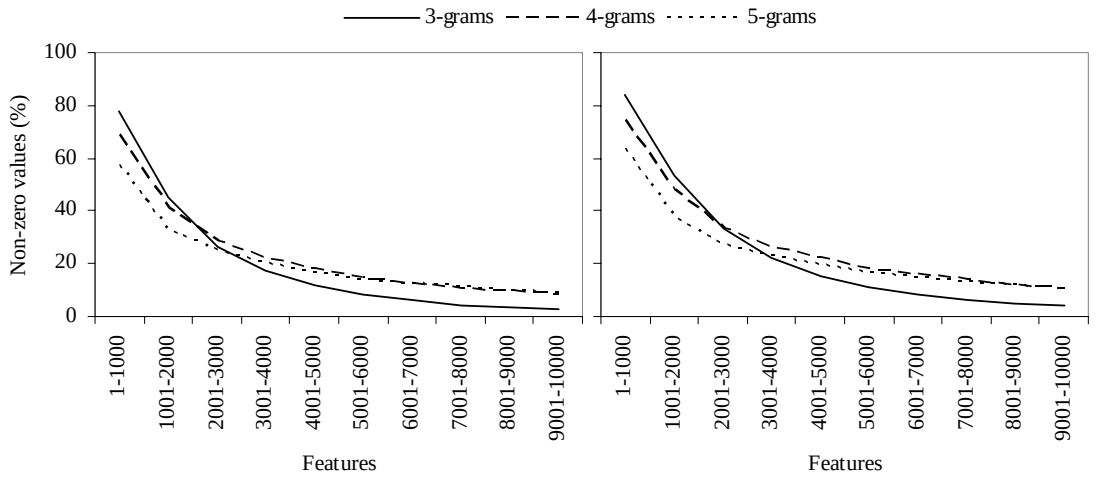


Figure 1. Average amount of non-zero attribute values per thousand of features for the training set of GA (left) and GB (right). Data sets of 3-grams, 4-grams and 5-grams are depicted.

	GA	GB
Avg. words per text	866.8	1,148.2
Authors	10	10
Texts per author	20	20
Texts per author in training set	10	10
Texts per author in test set	10	10
Reported Results (accuracy %)		
Stamatatos, *et al*., 2000 [18]	72	70
Peng, *et al*., 2003 [15]	74	90
Keselj, *et al*., 2003 [9]	**85**	**97**
Peng, *et al*., 2004 [16]	-	96

Table 1. The text corpora used in this study and reported accuracy results so far.

supplement B are written by scholars, writers, etc., rather than journalists.

Each corpus is divided into disjoint training and test parts of equal size in terms of texts per author (i.e., ten texts per author in the training set and ten texts per author in the test set for each group). Some brief information about these text corpora is summarized in Table 1. More detailed information can be found in [18]. Intuitively, for the GB it is easier to discriminate between the authors since the texts are more stylistically homogenous. In addition, GB's texts are significantly longer than GA's texts.

No linguistic preprocessing of the corpora is required for constructing the data sets for the current approach. The set of the d most frequent character *n*-grams (ordered by decreasing frequency of occurrence) of the training set is extracted, for a given character sequence length n. In the following experiments, character 3-grams, 4-grams, and 5-grams are examined while the feature set size (*d*) varies from 1,000 to 10,000. Then, each text is represented by the ordered vector of *d n*-gram frequencies, normalized over the total amount of text characters.

To illustrate the characteristics of these data sets, figure 1 depicts the average amount of non-zero attribute values per thousand of features for both GA and GB. As can be seen, the larger the feature set size, the sparser the resulting data. Moreover, shorter *n*-grams (i.e., 3-grams) tend to be less sparse for relatively low dimensional feature spaces (until 3,000 features). Of course, this can be explained by the fact that the complete set of 3-grams is much smaller than the complete set of 5-grams and the most frequent 3-grams are more likely to be found in every text in comparison to the most frequent 5-grams. On the other hand, beyond a certain level (around 3,000 most frequent *n*-grams) 3-grams are less likely to be found in a text in comparison to the corresponding 5-grams. Notice also that GA data sets are sparser in comparison to the corresponding GB data sets.

3.2 Setting the Baseline

The GA and GB corpora provide a reliable testing ground for author identification experiments since they comprise an adequate number of candidate authors, adequate number of test texts, and the authorship of each text is undisputed. For this reason, they were used to test several author identification approaches [9, 15, 16, 18] and the best reported results so far are shown in Table 1. Notice that the considerations about the difficulty of the two text corpora are reflected in the reported results since the classification accuracy for GB is much higher in comparison to GA.

As mentioned earlier, the approach described in [9] is also based on mere character *n*-grams, thus the comparison with the presented method is straightforward. Additionally, in order to test the proposed classification algorithm, a *Support Vector Machine* (SVM) model [21] was also built, since SVMs provide one of the best available solutions when dealing with high dimensional data.

4 RESULTS

The SVM and learning ensemble classification schemes were applied to both GA and GB. In particular, common kernel options that optimize the average performance of the models were selected (linear kernel, C=1). In particular, the exhaustive disjoint subspacing approach with minimal feature subset length (m=2) was followed. The base learner combination rule *mp* was used. For each text corpus, three different data sets were examined (3-grams, 4 grams, and 5-grams) with feature set size varying from 1,000 to 10,000 with a step of 1,000 *n*-grams. Table 2 shows the performance for both classification approaches on the test set of GA

Feat. set size	GA 3-grams SVM	GA 3-grams Ens.	GA 4-grams SVM	GA 4-grams Ens.	GA 5-grams SVM	GA 5-grams Ens.	GB 3-grams SVM	GB 3-grams Ens.	GB 4-grams SVM	GB 4-grams Ens.	GB 5-grams SVM	GB 5-grams Ens.
1,000	81	80	80	77	68	68	96	96	93	96	94	94
2,000	83	79	77	76	73	73	98	96	95	95	96	94
3,000	86	86	82	79	83	81	98	99	98	96	97	97
4,000	90	95	86	83	85	85	99	99	**100**	99	**100**	**100**
5,000	89	94	87	87	85	85	99	**100**	**100**	**100**	98	**100**
6,000	92	**96**	91	93	87	89	98	99	**100**	**100**	99	**100**
7,000	92	**96**	92	93	89	92	99	**100**	99	**100**	99	99
8,000	92	**96**	92	92	92	90	99	**100**	98	**100**	98	99
9,000	92	**96**	93	93	91	92	98	**100**	97	**100**	97	99
10,000	92	**96**	94	94	91	93	98	**100**	96	**100**	97	99

Table 2. Performance results on test set of both GA and GB for the support vector classifier and the learning ensemble. Classification accuracy (%) is indicated for different feature set size (amount of character *n*-grams) and types of features (3-grams, 4-grams, and 5-grams). Best achieved results are in boldface.

and GB. It is obvious that for GA it is more difficult to discriminate between the authors as compared with GB. Moreover, the best results for both approaches are much better than the best reported results for the same text corpora (see table 1). In more detail, in the best case, SVM achieves 94% and 100% classification accuracy for GA and GB, respectively, while the learning ensemble achieves 96% and 100% classification accuracy for GA and GB, respectively.

Notice that the performance of both approaches increases as the feature set size increases. Beyond a certain level (around 6,000 *n*-grams) the performance is either stabilized or slightly decreased (especially in the SVM models for the GB data sets). The ensemble model is superior of the SVM model in most cases with feature set size greater than 3,000. Therefore, it seems that the ensemble model is better able to handle high dimensional feature spaces. Additionally, in most cases 3-grams are better able to discriminate between the classes for both GA and GB. Recall that the 3-gram data sets are sparser beyond 6,000 features in comparison to 4-grams or 5-grams (see figure 1). Again, the ensemble model is superior for the 3-gram data sets and large feature set sizes. This indicates that the ensemble model can cope more effectively with sparse data.

4.1 Ensemble Diversity

A more detailed insight will illustrate why the ensemble model is so successful. The base classifiers that constitute the ensemble perform quite poorly when examined as individuals. Figure 2 depicts the base learner classification accuracy on the test data of GA and GB for the 3-gram data set. Random guess accuracy is indicated as well. As can be seen, the base classifiers are very poor predictors. Moreover, the predictions for GB are constantly more accurate than that of GA.

The key-factor for the success of the ensemble model is the extremely high diversity among the predictions of the base classifiers. Figure 3 shows the diversity, in terms of entropy, among the predictions of base classifiers on the test set of GB. Note that since the base classifiers are based on disjoint feature sets, the diversity is expected to be high. However, the level of entropy depicted in Figure 3 reaches 1.0, which means random error among the predictions. In words, the wrong predictions of the base classifiers are mutually cancelled.

Moreover, the diversity of the ensemble reaches its peak value at different size of feature set (and subsequently different amount of base classifiers), according to the data set. Thus, for the 3-gram data set, the diversity reaches its peak value at 5,000 features, while for the 4-gram and 5-gram data sets the diversity reaches its peak value at 7,000 and 8,000 features, respectively. Similar diversity curves can be obtained for the GA data sets. Notice that this decrease in diversity for the 3-gram data set of GA reflects in the performance of the corresponding ensemble models. Hence, the accuracy of the GA 3-gram ensemble model, shown in table 2, is not further improved for feature spaces greater than 5,000 features. However, despite this decrease in diversity, the classification accuracy does not drop (neither for GA nor GB).

4.2 Limited Training Texts

The training set size is a crucial factor in author identification since, in real world problems there is only a limited number of texts of undisputed authorship for each candidate author to be used as training data. For that reason, it is of vital importance for the classification method to require as limited training data as possible while maintaining a high level of accurate predictions on unseen cases.

To test the degree in which the SVM and the ensemble models are affected by the training set size, the experiment of the previous section was repeated based on reduced training sets. The SVM and the ensemble models were applied to both GA and GB using 50% (i.e., 5 texts per author) and 20% (i.e., 2 texts per author) of the original training sets. Data sets of 3-grams, 4-grams, and 5-grams of 10,000 features were examined. Table 3 shows the results of this experiment. Note that the test sets remain the same, thus, the results of Table 3 can be directly compared to Table 2. To illustrate further, the last line of table 3 indicates the performance of the models using the corresponding full-sized training sets (taken from Table 2).

In all cases the ensemble model performs better in comparison to SVM. In particular, for very limited training sets (20% of the original ones) the SVM model fails to maintain the previous classification accuracy. Interestingly, the performance of the ensemble model is not dramatically affected by reducing the training size. Actually, for the 3-gram data set of GB the classification accuracy remains at the top level using only 20% of the original training set, while for the corresponding GA data set the accuracy is competitive to the best reported results (see Table 1). In general, it seems that *n*-grams of short length (i.e., 3-grams) are better able to deal with limited training sets.

5 CONCLUSIONS

In this paper, an ensemble-based approach to the task of author identification was presented. Each text is represented as a vector of

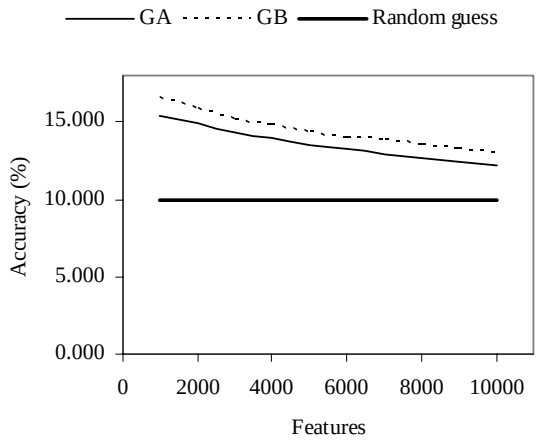


Figure 2. Classification accuracy (%) of the base classifiers for the 3-gram data set on GA and GB. Random guess accuracy is indicated as well.

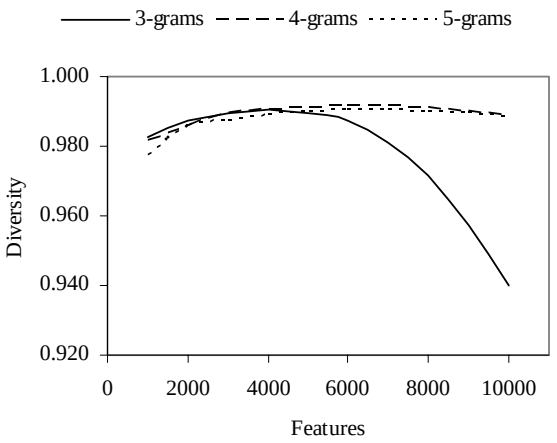


Figure 3. Diversity (in terms of entropy) of the base classifiers of the ensemble model for GB.

Train. set size	GA 3-grams SVM	GA 3-grams Ens.	GA 4-grams SVM	GA 4-grams Ens.	GA 5-grams SVM	GA 5-grams Ens.	GB 3-grams SVM	GB 3-grams Ens.	GB 4-grams SVM	GB 4-grams Ens.	GB 5-grams SVM	GB 5-grams Ens.
2	73	80	55	77	46	69	81	**100**	78	96	75	89
5	83	89	81	85	76	80	97	**100**	92	98	93	98
10	92	**96**	94	94	91	93	98	**100**	96	**100**	97	99

Table 3. Performance results on test set of both GA and GB for the support vector classifier and the learning ensemble. Classification accuracy (%) is indicated for different training set size (in texts per author) and types of features (3-grams, 4-grams, and 5-grams). In all cases, feature set size is 10,000. Best achieved results are in boldface.

frequencies of character *n*-grams. Such features require minimal text preprocessing and their extraction is a language-independent procedure. The ensemble-based approach of exhaustive disjoint subspacing was followed in order to handle such highly dimensional and sparse data. The application of this technique to two benchmark text corpora for author identification yields classification models with high accuracy, significantly higher than the best reported results for the same text corpora. First, this proves that character *n*-grams can successfully represent an author's style. Second, it demonstrates that the examined classification model can effectively cope with the author identification task.

The ensemble model proves to be significantly reliable when dealing with limited training set, a condition usually met in real-world author identification problems. Note also that the proposed technique does not require the use of a validation set for parameter tuning, minimizing the need for extra training texts. The success of the ensemble model is explained by the extremely high diversity among the predictions of the base classifiers. Previous studies have also shown that diversity alone can be used as a guide for constructing good ensembles [22]. The approach followed in this study ensures an extremely high level of diversity.

Special attention was paid on the combination of the predictions provided by the base classifiers. A scheme that combines the arithmetic and geometric mean is proposed. This scheme chooses the most likely class based on a compromise between high scores and low scores assigned to a class. The examined ensemble model is based on feature subsets of minimal length (m=2). This approach yields the highest number of base classifiers and provides the best experimental results. Moreover, it minimizes the effort to group features together in order to form feature subsets. Note that preliminary experiments with different subset lengths (m>2) indicated that the lower the feature subset length, the better (and more stable) the performance of the ensemble model.

In this study, features are paired at random. It has to be noted that repeated experiments with randomly paired features showed that the difference in performance is not statistically significant for feature sets including at least 3,000 features. However, a more sophisticated approach involving a search through the space of all the possible feature combinations [14] can also be examined. On the other hand, such an approach would require a validation set and a considerably greater training time cost.

As concerns the task of author identification, there are still open questions. In particular, limited text-length and imbalanced training set (i.e., unequal distribution of training texts over the authors) can affect the performance of the model. Moreover, open-class problems (i.e., the true author is not included in the candidate authors), another situation usually met in real-world problems, should be thoroughly examined as well.

REFERENCES

[1] Argamon, S., M. Saric, and S. Stein. 2003. Style Mining of Electronic Messages for Multiple Authorship Discrimination: First Results. In Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining: 475-480

[2] Bay S. 1998. Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets. In Proc. of the 15th International Conference on Machine Learning: 37-45

[3] Burrows, J.F. 1987. Word Patterns and Story Shapes: The Statistical Analysis of Narrative Style. Literary and Linguistic Computing, 2: 61-70.

[4] de Vel, O., A. Anderson, M. Corney, and G.M. Mohay. 2001. Mining E-mail Content for Author Identification Forensics. SIGMOD Record, 30(4): 55-64.
[5] Diederich, J., J. Kindermann, E. Leopold, and G. Paass. 2003. Authorship Attribution with Support Vector Machines. Applied Intelligence, 19(1/2): 109-123.
[6] Holmes, D. 1998. The Evolution of Stylometry in Humanities Scholarship. Literary and Linguistic Computing, 13(3): 111-117.
[7] Joachims, T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proc. of the 10th European Conference on Machine Learning.
[8] Juola, P. 2004. Ad-hoc Authorship Attribution Competition. In Proc. of the Joint Int. Conference ALLC/ACH 2004: 175-176.
[9] Keselj, V., F. Peng, N. Cercone, and C. Thomas. 2003. N-gram-based Author Profiles for Authorship Attribution. In Proc. of the Conference of the Pacific Association for Computational Linguistics.
[10] Koppel, M., and J. Schler. 2004. Authorship Verification as a One-Class Classification Problem. In Proc. of the Twenty-first Int. Conf. on Machine Learning.
[11] Kuncheva, L. & C. Whitaker. 2003. Measures of Diversity in Classifier Ensembles. Machine Learning, 51: 181-207.
[12] Lim, T., W. Loh, and Y. Shih. 2000. A Comparison of Prediction Accuracy, Complexity and Training Time of Thirty-three Old and New Classification Algorithms. Machine Learning, 40(3): 203-228.
[13] Mosteller, F. and D. Wallace. 1984. Applied Bayesian and Classical Inference: The Case of the Federalist Papers. Springer-Verlag, New York.
[14] Opitz, D., and J. Shavlik. 1999. A Genetic Algorithm Approach for Creating Neural Network Ensembles. In A. Sharkley (ed.) Combining Artificial Neural Nets: 79-99.
[15] Peng, F., D. Shuurmans, V. Keselj, and S. Wang. 2003. Language Independent Authorship Attribution Using Character Level Language Models. In Proc. of the 10th Conference of the European Chapter of the Association for Computational Linguistics.
[16] Peng, F., D. Shuurmans, and S. Wang. 2004. Augmenting Naive Bayes Classifiers with Statistical Language Models. Information Retrieval Journal, 7(1): 317-345.
[17] Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1): 1-47.
[18] Stamatatos, E., N. Fakotakis, and G. Kokkinakis. 2000. Automatic Text Categorisation in Terms of Genre and Author. Computational Linguistics, 26(4): 471-495.
[19] Tax, D., M. van Breukelen, R. Duin, and J. Kittler. 2000. Combining Multiple Classifiers by Averaging or by Multiplying? Pattern Recognition, 33: 1475-1485.
[20] van Halteren H. 2004. Linguistic Profiling for Author Recognition and Verification. In Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics: 199-206.
[21] Vapnik, V. 1995. The Nature of Statistical Learning Theory. Springer, New York.
[22] Zenobi, G., and P. Cunningham. 2001. Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error. In Proc. of 12th European Conference on Machine Learning: 576-587.

Syntax versus Semantics:
Analysis of Enriched Vector Space Models

Benno Stein[1] and **Sven Meyer zu Eissen**[1] and **Martin Potthast**[2]

Abstract. This paper presents a robust method for the construction of collection-specific document models. These document models are variants of the well-known vector space model, which relies on a process of selecting, modifying, and weighting index terms with respect to a given document collection. We improve the step of index term selection by applying statistical methods for concept identification. This approach is particularly suited for post-retrieval categorization and retrieval tasks in closed collections, which is typical for intranet search.

We compare our approach to "enriched" vector-space-based document models that employ knowledge of the underlying language in the form of external semantic concepts. Primary objective is to quantify the impact of a purely syntactic analysis in contrast to a semantic enrichment in the index construction step. As a by-product we provide an efficient and language-independent means for vector space model construction, whereas the resulting document models perform better than the standard vector space model.

Keywords vector space model, concept identification, semantic concepts, text categorization, evaluation measures

1 INTRODUCTION

Each text retrieval task that is automated by a computer relies on some kind of document model, which is an abstraction of the original document d. The document model must be tailored well with respect to the retrieval task in question: It determines the quality of the analysis, and—diametrically opposed—its computational complexity. Though its obvious simplicity the vector space model has shown great success in many text retrieval tasks [11; 12; 16; 15], and, the analysis of this paper uses this model as its starting point.

The standard vector space model abstracts a document d toward a vector $\mathbf{d}$ of weighted index terms. Each term t that is included in $\mathbf{d}$ derives from a term $\tau \in d$ by affix removal, which is necessary to map morphological variants of τ onto the same stem t. The respective term weights in $\mathbf{d}$ account for the different discriminative power of the original terms in d and are computed according to some frequency scheme. The main application of the vector space model is document similarity computation.

In this paper we focus on the index construction step and, in particular, on index term selection. Other concepts of the vector space model, such as the term weighting scheme or its disregard of word order are adopted.

[1] Faculty of Media, Media Systems.
Bauhaus University Weimar, 99421 Weimar, Germany
{benno.stein | sven.meyer-zu-eissen}@medien.uni-weimar.de
[2] Faculty of Computer Science.
Paderborn University, 33098 Paderborn, Germany

1.1 A Note on Semantics

We classify an index construction method as being semantic if it relies on additional domain knowledge, or if it exploits external information sources by means of some inference procedure, or both. Short documents may be similar to each other from the (semantic) viewpoint of a human reader, while the related instances of the vector space model do not reflect this fact because of the different words used. Index term enrichment can account for this by adding synonymous terms, hypernyms, hyponyms, or co-occurring terms [7].

Semantic approaches are oriented at the human understanding of language and text, and, as given in the case of ontological index term enrichment, they are computationally efficient. However, the application of the semantic approaches is problematic, if, for instance, the document language is unknown or if a document combines passages from several languages. Moreover, there are situations where semantic approaches can even impair the retrieval quality: Consider a document collection with specialized texts, then ontological index term enrichment will move the specific character of a text toward a more general understanding. As a consequence, the similarity of highly specialized text is diluted in favor of less specialized text—which compares to the effect of adding noise.

1.2 Contributions

We investigate variants of the vector space model with respect to their classification performance. Starting point is the standard vector space model where the step of index term selection is improved by a syntactic approach for concept identification; the resulting model is compared to semantically enriched vector space models. The syntactic concept identification approach is based on a collection-specific suffix tree analysis. In a nutshell, the paper's underlying question may be summarized as follows:

Can syntactically determined concepts keep up with a semantically motivated index term enrichment?

To answer this question we have set up a number of text categorization experiments with different clustering algorithms. Since these algorithms are susceptible to various side effects, we will also present results that rely on an objective similarity assessment statistic: the measure of expected density, $\overline{\rho}$. Perhaps the most interesting result may be anticipated: The positive effect of semantic index term enrichment, which has been reported by some authors in the past, could hardly be observed in our comprehensive analysis.

The remainder of the paper is organized as follows. Section 2 presents a taxonomy of index construction methods and outlines commonly used technology, and Section 3 reports on similarity analysis and unsupervised classification experiments.

2 INDEX CONSTRUCTION FOR DOCUMENT MODELS

This section organizes the current practice of index construction for vector space models. In particular, we review the concept of a document model and propose a classification scheme for both popular and specialized index construction principles.

A document d can be viewed under different aspects: layout, structural or logical setup, and semantics. A computer representation $\mathbf{d}$ of d may capture different portions of theses aspects. Note that $\mathbf{d}$ is designed purposefully, with respect to the structure of a formalized query, $\mathbf{q}$, and also with having a particular retrieval model in mind. A retrieval model, $\mathcal{R}$, provides the linguistic rationale for the model formation process behind the mapping $d \mapsto \mathbf{d}$. This mapping involves an inevitable simplification of d that should be

1. quantifiable,
2. useful with respect to the information need, and
3. tailored to $\mathbf{q}$, the formalized query.

The retrieval model $\mathcal{R}$ gives answers to these points, be it theoretically or empirically, and provides a concrete means, $\rho(\mathbf{q}, \mathbf{d})$, for quantifying the relevance between a formalized query $\mathbf{q}$ and a document's computer representation $\mathbf{d}$. Note that $\rho(\mathbf{q}, \mathbf{d})$ is often specified in the form of a similarity measure φ.

Together, the computer representation $\mathbf{d}$ along with the underlying retrieval model $\mathcal{R}$ form the document model; Figure 2 illustrates the connections.

Let D be a document collection and let T be the set of all terms that occur in D. The vector space model $\mathbf{d}$ of a document d is a vector of $|T|$ weights, each of which quantifying the "importance" of some index term in T with respect to d.[3] This quantification must be seen against the background that one is interested in a similarity function φ that maps from the vectors $\mathbf{d_1}$ and $\mathbf{d_2}$ of two documents d_1, d_2 into the interval $[0; 1]$ and that has the following property: If $\varphi(\mathbf{d_1}, \mathbf{d_2})$ is close to 1 then the documents d_1 and d_2 are similar; likewise, a value close to zero indicates a high dissimilarity. Note that document models and similarity functions determine each other: The vector space model and its variants are amenable to the cosine similarity (= normalized dot product) in first place, but can also be used in connection with Euclidean distance, overlap measures, or other distance concepts.

Under the vector space paradigm the document model construction process is determined in two dimensions: index construction and weight computation. In the following we will concentrate on the former dimension since this paper contributes right here. We have classified the index construction principles for vector space models in four main classes, which are shown in Figure 1.

[3] Note that, in effect, the vector space model is a computer representation of a the textual content of a document. However, in the literature the term "vector space model" is also understood as a retrieval model with a certain kind of relevance computation.

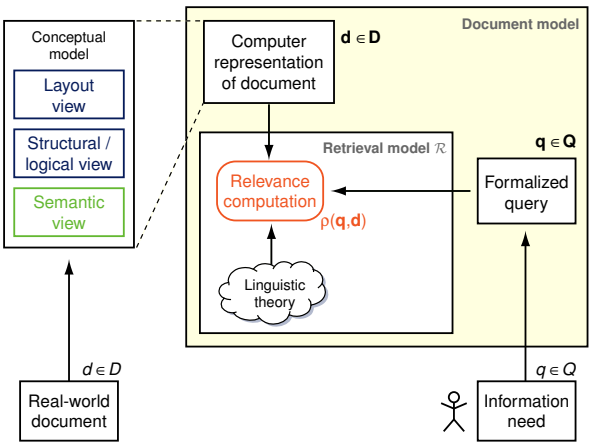


Figure 2. In the end, an information need q is satisfied by a real-world document d. A computer support for this retrieval task requires an abstraction of q and d towards $\mathbf{q}$ and $\mathbf{d}$. The rationale for this abstraction comes from a linguistic theory and is operationalized by a retrieval model $\mathcal{R}$.

Index Term Selection. Selection methods further divide into inclusion and exclusion methods. An important exclusion method is stopword removal: Common words, such as prepositions or conjunctions, introduce noise and provide no discriminating similarity information; they are usually discarded from the index set. However, there are special purpose models (e. g. for text genre identification) that rely on stopword features [13; 9].

The standard vector space model does not apply an inclusion method but simply takes the entire set T without stopwords. More advanced vector space models use also n-grams, i. e., continuous sequences of n words, $n \leq 4$, which occur in the documents of D. Since the usage of n-grams entails the risk of introducing noise, not all n-grams should be added but threshold-based selection methods be applied, which rely on the information gain or a similar statistic [6].

Index Term Modification. Most term modification methods aim at generalization. A common problem in this connection is the mapping of morphologically different words that embody the same concept onto the same index term. So-called stemming algorithms apply here; their goal is to find canonical forms for inflected or derived words, e. g. for declined nouns or conjugated verbs. Since the "unification" of words with respect to gender, number, time, and case is a language-specific issue, rule-based stemming algorithms require the development of specialized rule sets for each language. Recall that

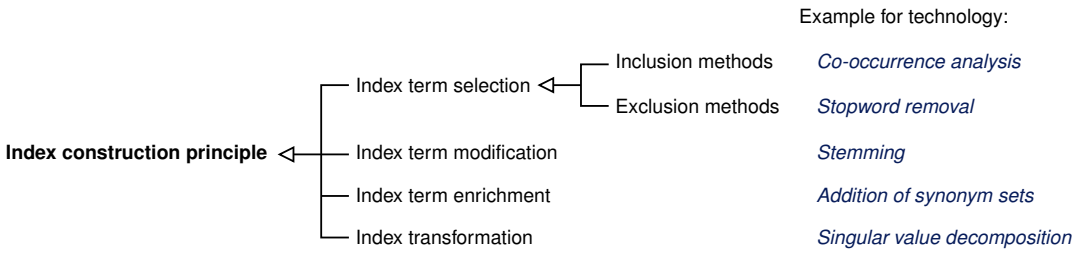


Figure 1. A taxonomy of index construction principles for vector space models.

the application of language-specific rule sets requires the problem of language detection both in unilingual and multilingual documents to be solved.

Index Term Enrichment. We classify a method as term enriching, if it introduces terms *not* found in T. By nature, meaningful index term enrichment must be semantically motivated and exploit linguistic knowledge. A standard approach is the—possibly transitive—extension of T by synonyms, hypernyms, hyponyms, and co-occurring terms. The extension shall alleviate the problem of different writing styles, or of vocabulary variations observed in very small document snippets as they are returned from search engines.

Note that these methods are not employed to address the problem of polysemy, since the required in-depth analysis of the term context is computationally too expensive for many similarity search applications.

Index Transformation. In contrast to the construction methods mentioned before, transformation methods operate on all document vectors of a collection D at the same time by analyzing the term-document matrix, A. A popular index transformation method is latent semantic indexing (LSI), which uses a singular value decomposition of A in order to improve query rankings and similarity computations [2; 1; 8]. For this purpose, the document vectors are projected into a low-dimensional space that is spanned by the eigenvectors that belong to the largest singular values of the decomposition of A.

2.1 Discussion

Index terms that consist of a single word can be found by a skillful analysis of prefix frequency and prefix length. This idea can be extended to the identification of compound word concepts in written text. If continuous sequences of n words occur significantly often, then it is likely that these words form a concept. Put another way, concept detection reduces to the identification of frequent n-grams.

n-grams as a replacement for index term enrichment has been analyzed by several authors in the past, with moderate success only [6]. We explain the disappointing results with noise effects, which dominate the positive impact of few additional concepts: Most authors apply a strategy of "complete extension"; i. e., they add all 2-grams and 3-grams to the index vector. However, when analyzing the frequency distribution of n-grams, it becomes clear that only a small fraction of all compound word sequences is statistically relevant.

The advantages of syntactical (statistical) methods for index construction can be summarized as follows:

1. language independence
2. robustness with respect to multi-lingual documents
3. tailored indexes for retrieval tasks on closed collections

An obvious disadvantage may be the necessary statistical mass: Syntactical index construction cannot work if only few, very small document snippets are involved. This problem is also investigated in the next section, where the development of the index quality is compared against the underlying collection size.

As an aside, statistical stemming and the detection of compound word concepts are essentially the same—the level of granularity makes the difference: Stemming means frequency analysis at the level of characters; likewise, the identification of concepts means frequency analysis at the level of words.

3 ANALYSIS OF ENRICHED VECTOR SPACE MODELS

Existing reports on the impact of index term selection and index term enrichment are contradictory [4; 5; 7], and not all of the published performance improvements could be reproduced [6]. Most of this research analyzes the effects of a modified vector space model on typical information retrieval tasks, such as document clustering or query answering.

Note that clustering results that have been obtained by employing the same cluster algorithm under different document models may tell us two things: (*i*) whether one document model captures more of the "gist" of the original document d than another model, and, (*ii*) whether the cluster algorithm is able to take advantage of this added value.

A cluster algorithm's performance depends on various parameters, such as the cluster number, its randomized start configuration, or preset similarity thresholds, etc., which renders a comparison difficult. Moreover, there is the prevalently observed effect that different cluster algorithms behave differently sensitive to document model "improvements". From an analysis point of view the following questions arise:

1. Which cluster algorithm shall define the baseline for a comparison (the best for the dataset, the most commonly used, the simplest)?
2. Given several clustering results obtained by the same cluster algorithm, which result can be regarded as meaningful (the best, the worst, the average)?

Especially to the second point less attention is paid in current research: Common practice is to select the best result compared to a given reference classification, e. g. by maximizing the F-Measure value—ignoring that such a combined usage of unsupervised/supervised methods is far away from reality.[4]

An objective way to rank different document models is to compare their ability to *capture the intrinsic similarity relations* of a given collection D. Basic idea is the construction of a similarity graph, measuring its conformance to a reference classification, and analyzing the improvement or decline of this conformance under some document model. Exactly this is operationalized in form of the $\overline{\rho}$-measure that is introduced below; it enables one to evaluate differences in the similarity concepts of alternative document models without being dependent on a cluster algorithm.[5]

Hence, the performance analyses presented in this section comprise two types of analyses: (*i*) Experiments that, based on $\overline{\rho}$, quantify objective improvements or declines of a document model, (*ii*) experiments that, based on the F-Measure, quantify the effects of a document model onto different cluster algorithms.

3.1 A Measure of Expected Density: $\overline{\rho}$

As before let $D = \{d_1, \ldots, d_n\}$ be a document collection whose corresponding computer representations are denoted as $\mathbf{d}_1, \ldots, \mathbf{d}_n$. A similarity graph $G = \langle V, E, \varphi \rangle$ for D is a graph where a node in V represents a document and an edge $(d_i, d_j) \in E$ is weighted with the similarity $\varphi(\mathbf{d}_i, \mathbf{d}_j)$.

A graph $G = \langle V, E, w \rangle$ is called sparse if $|E| = \mathcal{O}(|V|)$; it is called dense if $|E| = \mathcal{O}(|V|^2)$. Put another way, we can compute the density θ of a graph from the equation $|E| = |V|^\theta$. With

[4] This issue is addressed in [14].

[5] The $\overline{\rho}$-measure was originally introduced in [14], as an alternative for the Davies-Bouldin-Index and the Dunn-Index, in order to evaluate the quality of cluster algorithms for text retrieval applications.

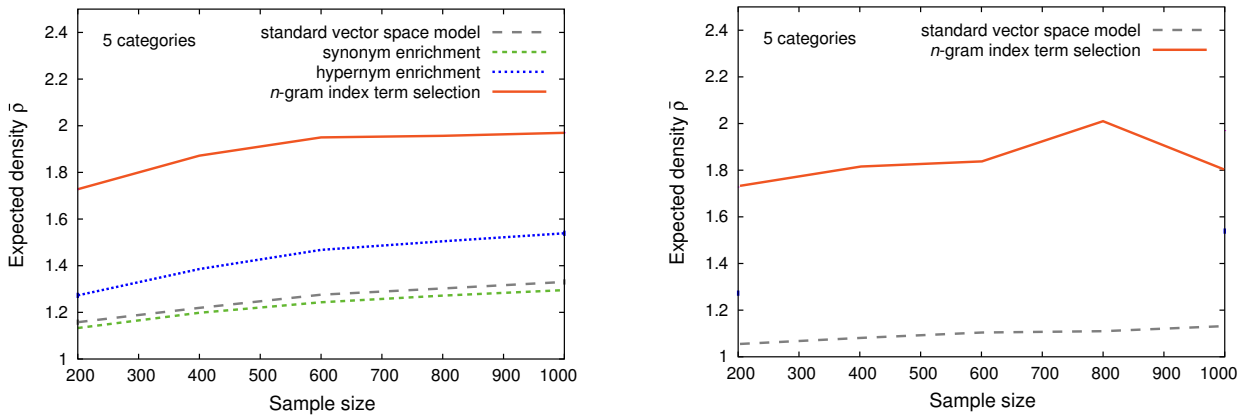


Figure 3. Comparison of the standard vector space model, two semantically enriched models (synonym, hypernym), and a vector space model with syntactically identified concepts (n-gram) in two languages: The left-hand graph illustrates the development of $\overline{\rho}$ depending on the collection size for an English document collection; the right-hand graph compares the n-gram vector space model to the standard model for a German document collection.

$w(G) := |V| + \sum_{e \in E} w(e)$, this relation extends naturally to weighted graphs:[6]

$$w(G) = |V|^{\theta} \quad \Leftrightarrow \quad \theta = \frac{\ln\big(w(G)\big)}{\ln\big(|V|\big)}$$

Obviously, θ can be used to compare the density of each induced subgraph $G' = \langle V', E', w' \rangle$ of G to the density of G: G' is sparse (dense) compared to G if the quotient $w(G')/(|V'|^{\theta})$ is smaller (larger) than 1. This consideration provides a key to quantify a document model's ability to capture the intrinsic similarity relations of G, and hence, of the underlying collection.

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be an exclusive categorization of D in k distinct categories, that is to say, $C_i, C_j \subseteq D$ with $C_i \cap C_j = \emptyset$ and $\cup_{i=1}^{k} C_i = D$, and let $G_i = \langle V_i, E_i, \varphi \rangle$ be the induced subgraph of G with respect to category C_i. Then the expected density of $\mathcal{C}$ is defined as follows.

$$\overline{\rho}(\mathcal{C}) = \sum_{i=1}^{k} \frac{|V_i|}{|V|} \cdot \frac{w(G_i)}{|V_i|^{\theta}}, \quad \text{where} \quad |V|^{\theta} = w(G)$$

Since the edge weights resemble the similarity of the documents associated with V, a higher value of $\overline{\rho}$ indicates a better modeling of a collection's similarity relations.

3.2 Syntax versus Semantics: Variants of the Vector Space Model

Aside from the standard vector space model our analysis compares the following three vector space model variants:

1. *Syntactic Term Selection.* Within this variant the index term selection step also considers syntactically identified concepts, i. e., 2-grams, 3-grams, and 4-grams. To identify the significant n-grams the document collection D is inserted into a suffix tree and a statistical successor variety analysis is applied. The operationalized principle behind this analysis is the peak-and-plateau method [5], for which we have developed a refinement in our working group.
2. *Semantic Synonym Enrichment.* Within this variant of semantic term enrichment the so-called synsets from Wordnet for nouns are added [3]; this procedure has been reported to work well for categorization tasks [7]. Note that adding synonyms to all index terms of a document vector will introduce a lot of noise, and hence only the top-ranked 10% of the index terms (respecting the employed term weighting scheme) are selected for enrichment.
3. *Semantic Hypernym Enrichment.* This variant of semantic term enrichment relies also on Wordnet: a sequence of up to four consecutive hypernyms is substituted for each noun. The rationale is as follows. Documents dealing with closely related—but still different—topics often contain terms which derive from a single hypernym representing their common category. The enrichment proposed here yields a stronger similarity between such documents without generalizing too much.

Index term weighting of both unigrams and n-grams follows the $tf \cdot idf$-scheme; stopwords are not indexed and unigram stemming is done according to Porter's algorithm.

Discussion. The resulting graphs in Figure 3 as well as the comparison in Table 1 show that the syntactic approach outperforms both semantic approaches. From the semantic variants only the semantic hypernym enrichment is above the baseline; note that this happens even if a large number synsets is added. We explain the results as follows: Index terms with a high term weight typically belong to a special vocabulary, and, from a semantic point of view, they are used deliberately so that adding their synsets will tend to *decrease* their importance. Likewise, adding the synsets of low-weighted terms has no effect other than adding noise since the importance of these terms will be *increased without a true rationale*.

Vector space model variant	F-min	F-max	F-av.
	(sample size 1000, 10 categories)		
standard vector space model	—baseline—		
synonym enrichment	-8%	+4%	-2%
hypernym enrichment	+5%	+12%	+3%
n-gram index term selection	+15%	+6%	+8%

Table 1. The table shows the improvements of the averaged F-Measure values that were achieved with the cluster algorithms k-means and MajorClust for the investigated variants of the vector space model.

[6] $w(G)$ denotes the total edge weight of G plus the number of nodes, $|V|$, which serves as adjustment term for graphs with edge weights in $[0; 1]$.

3.3 Test Corpus and Sample Formation

Experiments have been conducted with samples from RCV1, a short hand for "Reuters Corpus Volume 1" [10], as well as with documents from German newsgroup postings.

RCV1 is a document collection that was published by the Reuters Corporation for research purposes. It contains more than 800,000 documents each of which consisting of a few hundred up to several thousands words. The documents are tagged with meta information like category (also called topic), geographic region, or industry sector. There are 103 different categories, which are arranged within a hierarchy of the four top level categories "Government, Social", "Economics", "Markets", and "Corporate, Industrial". Each of the top level categories defines the root of a tree of sub-categories, where each child node fine grains the information given by its parent. Note that a document d can be assigned to several categories $c_1, \ldots, c_p$, and that d does also belong to all ancestor categories of some category c_i.

Within our experiments two documents d_i, d_j are considered to belong to the same category if they share the same top level category c_t and the same most specific category c_s. Moreover, the test sets are constructed in such a way that there is no document d_i whose most specific category c_s is an ancestor of the most specific category of some other document d_j.

The samples were formed as follows: For the analysis of the intrinsic similarity relations based on $\overline{\rho}$, the sample sizes ranged from 200 to 1000 documents taken from 5 categories. For the analysis of the categorization experiments, based on cluster algorithms and evaluated with the F-Measure, the sample sizes were 1000 documents taken from 10 categories.[7]

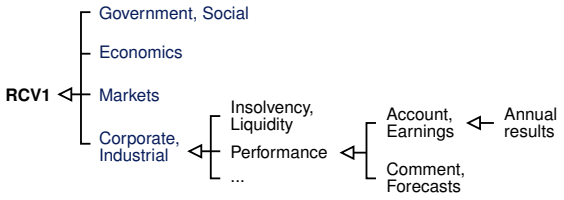


Figure 4. Category organization of the RCV1 corpus showing the four top level categories from which "Corporate, Industrial" is further refined.

4 SUMMARY

This paper provided a comparison of syntactical and semantic methods for the construction of vector space models; the special focus was index term selection. Interestingly, little attention has been paid to the mentioned syntactical methods in connection with text retrieval tasks. Following results of our paper shall be emphasized:

- With syntactically identified concepts significant improvements can be achieved for categorization tasks.
- The benefit of semantic term enrichment is generally overestimated.
- The $\overline{\rho}$-measure provides an "algorithm-neutral" approach to analyze the similarity knowledge contained in document models.

[7] To make our analysis results reproducible for other researchers, meta information files that describe the compiled test collections have been recorded; they are available upon request.

Note that the last point may be interesting to develop accepted benchmarks to compare research efforts related to document models or similarity measures.

Though syntactical analyses must not be seen as a cure-all for the index construction of vector space models, they provide advantages over semantic methods, such as language independence, robustness, and tailored index sets. With respect to several retrieval tasks they can keep up with semantic methods—however, our results give no room for an over-simplification: Both paradigms have the potential to outperform the other.

References

[1] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien, 'Using Linear Algebra for Intelligent Information Retrieval', Technical Report UT-CS-94-270, Computer Science Department, (dec 1994).

[2] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, 'Indexing by Latent Semantic Analysis', *Journal of the American Society of Information Science*, **41**(6), 391–407, (1990).

[3] Christiane Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[4] W. B. Frakes, 'Term conflation for information retrieval', in *SIGIR '84: Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 383–389, Swinton, UK, (1984). British Computer Society.

[5] W. B. Frakes and Ricardo Baeza-Yates, *Information retrieval: Data Structures and Algorithms*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[6] Johannes Fürnkranz, 'A Study Using n-gram Features for Text Categorization', Technical report, Austrian Institute for Artificial Intelligence, (1998). Technical Report OEFAI-TR-9830.

[7] A. Hotho, S. Staab, and G. Stumme, 'Wordnet improves text document clustering', in *Proceedings of the SIGIR Semantic Web Workshop*, (2003).

[8] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala, 'Latent semantic indexing: a probabilistic analysis', in *PODS '98: Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 159–168, New York, NY, USA, (1998). ACM Press.

[9] Andreas Rauber and Alexander Müller-Kögler, 'Integrating automatic genre analysis into digital libraries', in *ACM/IEEE Joint Conference on Digital Libraries*, pp. 1–10, (2001).

[10] T.G. Rose, M. Stevenson, and M. Whitehead, 'The Reuters Corpus Volume 1 - From Yesterday's News to Tomorrow's Language Resources', in *Proceedings of the Third International Conference on Language Resources and Evaluation*, (2002).

[11] G. Salton and M. E. Lesk, 'Computer Evaluation of Indexing and Text Processing', *ACM*, **15**(1), 8–36, (January 1968).

[12] Karen Sparck-Jones, 'A statistical interpretation of term specificity and its application in retrieval', *Journal of Documentation*, **28**, 11–21, (1972).

[13] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, 'Text genre detection using common word frequencies', in *Proceedings of the 18th Int. Conference on Computational Linguistics*, Saarbrücken, Germany, (2000).

[14] Benno Stein, Sven Meyer zu Eißen, and Frank Wißbrock, 'On Cluster Validity and the Information Need of Users', in *Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA 03),*

Benalmádena, Spain, ed., M. H. Hanza, pp. 216–221, Anaheim, Calgary, Zurich, (September 2003). ACTA Press.
[15] Michael Steinbach, George Karypis, and Vipin Kumar, 'A comparison of document clustering techniques', Technical Report 00-034, Department of Computer Science and Egineering, University of Minnesota, (2000).
[16] Yiming Yang and Jan O. Pedersen, 'A comparative study on feature selection in text categorization', in *Proceedings of ICML-97, 14th International Conference on Machine Learning*, ed., Douglas H. Fisher, pp. 412–420, Nashville, US, (1997). Morgan Kaufmann Publishers, San Francisco, US.

Challenges in Extracting Terminology from Modern Greek Texts

Aristomenis Thanopoulos and **Katia Kermanidis and Nikos Fakotakis**[1]

Abstract. This paper describes the automatic extraction of economic terminology from Modern Greek texts as a first step towards creating an ontological thesaurus of economic concepts. Unlike previous approaches, the domain-specific corpus utilized is varying in genre, and therefore rich in vocabulary and linguistic structure, while the pre-processing level is relatively low (basic morphological tagging, the detection of elementary, non-overlapping chunks) and fully automatic. The idiosyncratic properties of Modern Greek noun phrases are taken into account: the freedom in word ordering, the richness in morphology. Also, the peculiarity of the available corpora is dealt with: the large size of the economic compared to the balanced corpus. A combination of statistical filters (relative frequency ratios and log likelihood) and smoothing is employed in order to deal with the afore-mentioned challenges when filtering out non-terms.

1 INTRODUCTION

Terms are the linguistic expression of concepts. Domain-specific terms capture the knowledge of a given domain and reflect it in the form of words that are commonly acceptable by the members of the domain community, enabling the latter to interact and exchange information. In contrast to the use of static dictionaries, acquiring terminology automatically from domain texts leads to a list of extracted terms that may be dynamically updated and ranked according to usage. Term extraction is a first step towards acquiring a domain ontology. An ontology is a thesaurus that provides the relationships among the terms, and sorts them in a hierarchical structure, based on their semantic specificity and their properties.

Several methods have been employed for the extraction of domain terms. Regarding the linguistic pre-processing of the text corpora, approaches vary from simple tokenization and part-of-speech tagging ([1],[2]), to the use of shallow parsers and higher-level linguistic processors ([4],[8]). The latter aim at identifying syntactic patterns, like noun phrases, and their structure (e.g. head-modifier), in order to rule out tokens that are grammatically impossible to constitute terms (e.g. adverbs, verbs, pronouns, articles, etc).

Regarding the statistical filters, that have been employed in previous work to filter out non-terms , they also vary. Using corpus comparison, the techniques try to identify words/phrases that present a different statistical behavior in the corpus of the target domain, compared to their behavior in the rest of the corpora. Such words/phrases are considered to be terms of the domain in question. In the most simple case, the observed frequencies of the candidate terms are compared ([1]). Kilgarriff in [6] experiments with various other metrics, like the χ^2 score, the t-test, mutual information, the Mann-Whitney rank test, the Log Likelihood, Fisher's exact test and the TF.IDF (term frequency-inverse document frequency). Frantzi et al. in [2] present a metric that combines statistical (frequencies of compound terms and their nested sub-terms) and linguistic (context words are assigned a weight of importance) information.

In this paper we present the first phase of the ongoing work towards the creation of an ontology hierarchy of economic concepts. This phase includes the extraction of economic terms automatically from a Modern Greek phrase-analyzed corpus by corpora comparison in combination to applying a threshold to the relative frequency ratios.

An important aspect of the present approach is the stylistic nature of the domain-specific (economic) corpus. In most of the previous work, the domain corpus is to a large extent restricted in the vocabulary it contains and in the variety of syntactic structures it presents. Our economic corpus does not consist of syntactically standardized taglines of economic news. On the contrary, it presents a very rich variety in vocabulary, syntactic formulations, idiomatic expressions, sentence length, making the process of term extraction an interesting challenge.

In addition to this, the employed pre-processing tools (shallow phrase chunker) make use of limited resources (see section 2.2) and the question arises whether the resulting low-level information is sufficient to deal with the linguistic complexity of the corpus.

Another challenge that has been faced by the present work is the language itself. In Modern Greek the ordering of the constituents of a sentence or a phrase is loose and determined primarily by the rich morphology. As a result, the extraction of compound terms, as well as the identification of nested terms, are not straightforward and cannot be treated as cases of simple string concatenation, as in English. Section 2.3 describes an approach for extracting the counts of candidate terms, which takes into account the freedom in word ordering.

Finally, a peculiar trait of the current work is the corpora that are available to us. While the economic corpus is sufficiently large, the balanced corpus is relatively small. As a result, the terms (especially bi-grams) that occur in both corpora are few, while many valid terms appear in the domain specific corpus alone. This makes it impossible to use the traditional methodology of corpora comparison alone (that presupposes the appearance of a candidate term in both corpora) in order to filter out non-terms. A smoothing technique is applied to overcome this problem, which is described in section 3.

[1] Wire Communications Laboratory, University of Patras, Greece. Email: {aristom, kerman, fakotaki}@wcl.ee.upatras.gr

2 LINGUISTIC PROCESSING

A set of linguistic processing tools have been employed in order to parse the textual corpora. The first goal is to detect nouns (e.g. *τράπεζα* - bank), nominal compounds (*αύξηση κεφαλαίου* - capital increase) and named entities (*Τράπεζα της Ελλάδος* - Bank of Greece). All the above structures appear in the noun and prepositional phrases in a sentence. These types of phrases need to be detected, non-content words that appear in them have to be disregarded, and the candidate economic terms need to be formed. This process is described in detail in the rest of this section.

2.1 Modern Greek

Regarding the properties of the language that are strongly related to the current task, it has to be taken into account that Modern Greek is highly inflectional. The rich morphology allows for a larger degree of freedom in the ordering of the constituents of a phrase (headword and modifiers), compared to other languages such as English or German. More specifically, modifiers like adjectives, numerals and pronouns may precede or follow the head noun.

Another common property of noun phrases is the presence of nominal modifiers in the genitive case that denote possession, quality, quantity or origin. They are nouns and usually follow the head noun they modify.

The following two examples show the afore-mentioned freedom. The two phrases have exactly the same meaning (*bank account*). The first phrase is an adjective-noun construction, while the second is a noun-genitive modifier construction.

τραπεζικός λογαριασμός	$\text{bank}_{[\text{ADJECTIVE}]}$ $\text{account}_{[\text{NOUN}]}$
λογαριασμός τράπεζας	$\text{account}_{[\text{NOUN}]}$ $\text{bank}_{[\text{NOUN-GENITIVE}]}$

2.2 Corpora and processing tools

The corpora used in our experiments were:

1. The ILSP/ELEFTHEROTYPIA ([3]) and ESPRIT 860 ([9]) Corpora (a total of 300,000 words). Both these corpora are balanced and manually annotated with complete morphological information. Further (phrase structure) information is obtained automatically.

2. The DELOS Corpus, [5], is a collection of economic domain texts of approximately five million words and of varying genre. It has been automatically annotated from the ground up. Morphological tagging on DELOS was performed by the analyzer of [10]. Accuracy in part-of-speech and case tagging reaches 98% and 94% accuracy respectively. Further (phrase structure) information is again obtained automatically.

All of the above corpora (including DELOS) are collections of newspaper and journal articles. More specifically, regarding DELOS, the collection consists of texts taken from the financial newspaper EXPRESS, reports from the Foundation for Economic and Industrial Research, research papers from the Athens University of Economics and several reports from the Bank of Greece. The documents are of varying genre like press reportage, news, articles, interviews and scientific studies and cover all the basic areas of the economic domain, i.e. microeconomics, macroeconomics, international economics, finance, business administration, economic history, economic law, public economics etc. Therefore, it presents a richness in vocabulary, in linguistic structure, in the use of idiomatic expressions and colloquialisms, which is not encountered in the highly domain- and language-restricted texts used normally for term extraction (e.g. medical records, technical articles, tourist site descriptions). To indicate the linguistic complexity of the corpus, we mention that the length of noun phrases varies from 1 to 53 word tokens.

All the corpora have been phrase-analyzed by the chunker described in detail in [11]. Noun, verb, prepositional, adverbial phrases and conjunctions are detected via multi-pass parsing. From the above phrases, noun and prepositional phrases only are taken into account for the present task, as they are the only types of phrases that may include terms. Regarding the phrases of interest, precision and recall reach 85.6% and 94.5% for noun phrases, and 99.1% and 93.9% for prepositional phrases respectively. The robustness of the chunker and its independence on extravagant information makes it suitable to deal with a style-varying and complicated in linguistic structure corpus like DELOS.

It should be noted that phrases are non-overlapping. Embedded phrased are flatly split into distinct phrases. Nominal modifiers in the genitive case are included in the same phrase with the noun they modify; nouns joined by a coordinating conjunction are grouped into one phrase. The chunker identifies basic phrase constructions during the first passes (e.g. adjective-nouns, article nouns), and combines smaller phrases into longer ones in later passes (e.g. coordination, inclusion of genitive modifiers, compound phrases). As a result, named entities, proper nouns, compound nominal constructions are identified during chunking among the rest of the noun phrases.

The most significant sources of error during the automatic chunking process, which also affect the performance of the term extraction process, are:

1. Excessive phrase cut-up, usually due to erroneous part-of-speech tagging of a word (the word *πλήρες* - full - in the following example is erroneously tagged as a noun and not as an adjective)

NP[*Το πλήρες****]*** ***NP[****κείμενο της ανακοίνωσης****]*** instead of

NP[*Το πλήρες κείμενο της ανακοίνωσης****]***

*(****NP[****the full text of the announcement****])*

2. Erroneous NP tagging (unidentifiable adverbs, like *όντως* – in fact – in the following example, are marked as nouns)

NP[*όντως****]*** instead of ***ADV[****όντως****]***

In order to detect simple phrases inside larger coordination constructions, we applied the following simple empirical grammar to every noun and prepositional phrase extracted by the chunker. The grammar, which directly identifies conjunctive expressions and produces a list of simple noun phrases, employs the following rules:

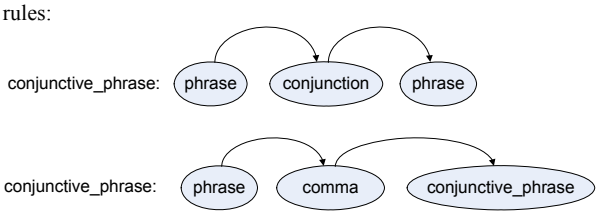


Figure 1. The rules for splitting coordinated phrases.

2.3 Candidate terms

As mentioned before, the noun and prepositional phrases of the two corpora are selected, as only these phrases are likely to contain

terms. Words of no semantic content (i.e. introductory articles, adverbs, prepositions, punctuation marks and symbols) are removed from the phrases.

Coordination schemes are detected within the phrases, and the latter are split into smaller phrases respectively according to the grammar depicted in Figure 1. The occurrences of words and N-grams, pure as well as nested, are counted. Longer candidate terms are split into smaller units (tri-grams into bi-grams and uni-grams, bi-grams into uni-grams).

Regarding the bi-grams, in order to overcome the freedom in the word ordering, as discussed in section 2.1, we considered bi-gram *A B* (*A* and *B* being the two lemmata forming the bi-gram) to be identical to bi-gram *B A*, if the bi-gram is not a named entity. Their joint count in the corpora is calculated and taken into account. The resulting uni-grams and bi-grams are the candidate terms. The candidate term counts in the corpora are then used in the statistical filters described in the next section.

Figure 2 shows the count calculation for the nested candidate terms. The two tri-grams, *A B C* and *B C D* occur in a corpus three and four times respectively. The accumulative counts of the nested terms are shown in parentheses.

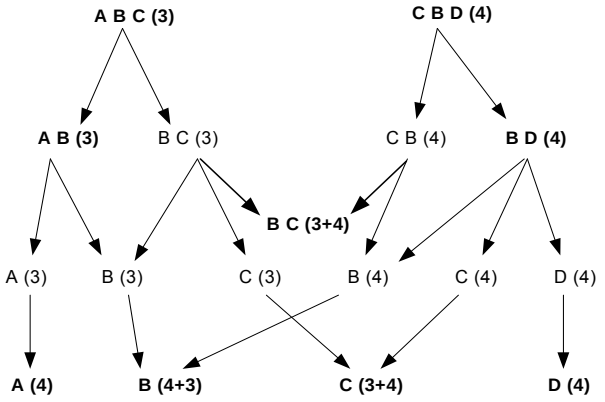


Figure 2. Calculation of n-gram frequencies, given the phrase-chunked corpus. The finally extracted n-gram frequencies are indicated in bold.

3 TERM FILTERING

In this section we describe the statistical filters that have been used to filter out non-terms. With *D* we denote Delos and with *B* the balanced corpus. As a first step, the occurrences of each candidate term *w* ($c_w(D)$ and $c_w(B)$) are counted in the two corpora separately.

A particularity of the present work is that, unlike in most previous approaches to term extraction, the domain-specific corpus available to us is quite large compared to the balanced corpus. As a result, several terms that appear in DELOS do not appear in the balanced corpus, making it impossible for the LLR statistic to detect them. In other words, these terms cannot be identified by traditional corpora comparison.

In order to deal with this phenomenon, we applied a smoothing technique to take into account terms that do not appear in the balanced corpus. More specifically, we applied Lidstone's law ([7]) to our candidate terms, i.e. we augmented each candidate term count by a value of λ=0.5 in both corpora. Thereby, terms that actually do not appear in the balanced corpus at all, end up having $c_w(B)=0.5$.This value was chosen for λ because, due to the small size of the balanced corpus, the probability of coming across a previously unseen word is significant.

Filtering was then performed in two stages: First the relative frequencies are calculated for each candidate term *w*, as

$$RF_w = f_w(D)/f_w(B), \quad (1)$$

$$f_w(D) = c_w(D)/N \quad (2)$$

$$f_w(B) = c_w(B)/M \quad (3)$$

N and *M* denote the counts of all candidate terms in *D* and *B* respectively.

In the next step, for those candidate terms that present an $RF_w>1$, LLR is calculated (according to the formula of [6]) as

$$\begin{aligned} LLR_w = 2\cdot(&c_w(D)\cdot\log(c_w(D)) + c_w(B)\cdot\log(c_w(B)) + \\ &(N-c_w(D))\cdot\log(N-c_w(D)) + (M-c_w(B))\cdot\log(M-c_w(B)) - \\ &(c_w(D)+c_w(B))\cdot\log(c_w(D)+c_w(B)) - M\cdot\log M - N\log N - \\ &(N+M-c_w(D)-c_w(B))\cdot\log(N+M-c_w(D)-c_w(B)) + \\ &(N+M)\cdot\log(N+M)\ ) \end{aligned} \quad (4)$$

The LLR metric detects how surprising (or not) it is for a candidate term to appear in DELOS or in the balanced corpus (compared to its expected appearance count), and therefore constitute an economic domain term (or not). Unlike other statistics (like the χ^2 and mutual information), it is an accurate measure even for rare candidate terms, and for this reason it was selected for the present task. It is asymptotically χ^2 distributed. So, for one degree of freedom, candidate terms that present an LLR value greater than 7.88 (critical value) can be considered as valid terms with a confidence level of 0.005.

4 EXPERIMENTAL RESULTS

The final list of extracted terms was evaluated by a group of three experts in economics and finance. The evaluators were in constant contact to agree upon ambiguous cases of terms. The most important factor for this ambiguity is the lack of context information, especially for uni-grams. In other words, there are several cases of words that may or may not be economic terms depending on the context in which they appear.

Table 1 lists a window from the list of the candidate terms, selected by chance. Their counts in both corpora are also shown (original counts, prior to smoothing), along with their RF value, and the tags that were given to them by the experts. These are terms with either RF<<1 or RF >>1, i.e. terms that present a significant difference between their frequencies in the two corpora, and so they vary from strongly economic (e.g. *tax-related*) to non-economic (*island*).

As the LLR threshold value decreases (the N-best number increases), the number of non-economic and mostly non-economic terms that enters into the N-best terms also increases causing the precision to drop.

The results cannot be easily compared to those of previous approaches, due to the many differences in resources and pre-processing. Merely as an indication, these results are comparable to the ones reported in [1] (73% to 86% precision, using a threshold on term frequencies in technical corpora on fiber optic networks, depending on the specific domain corpus and the size of the extracted list of candidate terms, which is similar to the list size in the current work).

Figure 3 shows the percentage of terms that have been correctly labeled as valid terms (y-axis) when taking into account the N-best labeled terms (x-axis) (i.e. for different LLR thresholds). This graph refers to terms that appear in both corpora and for which $RF_w>1$. *Strongly economic* are terms that are characteristic of the

Table 1. The 24 terms with the highest LLR scores along with their counts and their domain relevance.

word	translation	DILOS Freq.	IEL Freq.	Relative Freq. Ratio	LLR	Important to the Domain	Possibly Important to Domain	Unimportant to Domain
φορολογικός	tax-related	352	13	4,63	49,0	✓	-	-
π ρ α ώ	present	13	24	0,09	48,5	-	-	✓
γ λα ω	language	13	24	0,09	48,5	-	-	✓
αρ σ ερι τ ός	left, leftist	7	20	0,06	48,3	-	✓	-
εσωκομμ τικός	intra-party (political)	10	22	0,08	48,1	-	✓	-
διάλογος	dialog	131	68	0,33	47,4	-	-	✓
πετρ λαιοέ	oil (petrol)	213	3	12,14	47,2	✓	-	-
κ ρδο ε αφ	profitability	164	0	-	47,1	✓	-	-
πρ ψ βλ	prediction ε	283	8	6,05	46,9	✓	-	-
νη σ ί	island	14	24	0,10	46,8	-	-	✓
ά κ	anchorp	4	17	0,04	46,2	-	-	✓
γεν ι	yen	161	0	-	46,1	✓	-	-
σ ό τ χ	o target	821	64	2,19	46,1	✓	-	-
αστυνομία	police	45	38	0,20	46,0	-	✓	-
εργάτης	factory worker	3	16	0,03	45,9	-	✓	-
προοπτική	prospect	446	23	3,32	45,8	✓	-	-
OTE	HTO (company)	149	0	-	45,8	✓	-	-
σ υία μ	agreement	654	45	2,49	45,8	✓	-	-
γ ρ ανικ μ ός	German	238	5	8,14	45,7	-	✓	-
πολιτισμός	culture	31	32	0,17	45,6	-	✓	-
δουλειά	job, work	38	35	0,19	45,6	-	✓	-
διευθύνων	chief (executive)	199	3	11,43	45,6	✓	-	-
διο κ ητι τ ός	administrative	278	8	5,94	45,6	✓	-	-
ισ ιμ αο ί τ	currency	182	2	15,68	45,4	✓	-	-

domain and necessary for understanding domain texts. *Economic* are terms that function as economic within a context of this domain, but may also have a different meaning outside this domain. As regards to the aforementioned labeling, this category includes terms connected both directly and indirectly to the domain. *Mostly non-economic* are words that are connected to the specific domain only indirectly, or more general terms that normally appear outside the economic domain, but may carry an economic sense in certain limited cases. *Non-economic* are terms that never appear in an economic sense or can be related to the domain in any way. For example, referring to Table 1, "φορολογικός" ("tax" [adjective]) is considered as a strongly economic term, while "πολιτισμός" ("culture") is characterised as possibly important to the domain of economics, since it often involves a financial level.

Figure 4 shows the precision achieved for the terms appearing in both corpora that present an RF<1. It is an interesting graph to observe, in combination with Figure 3, as it shows how the method performs for the terms that are more frequent in the balanced corpus in comparison to DELOS.

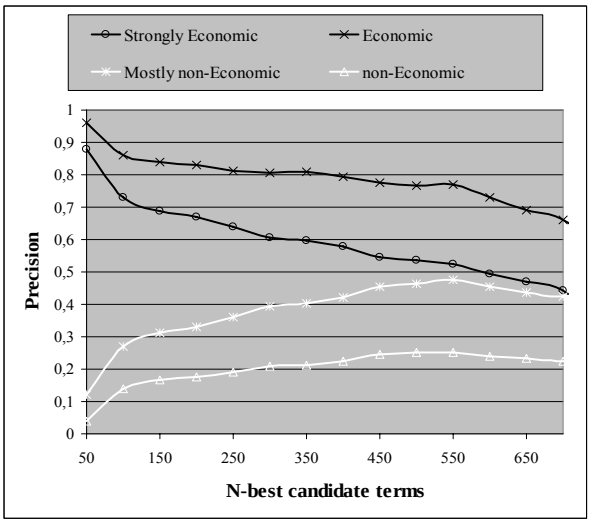


Figure 3. Precision (y-axis) for the N-best candidate terms (x-axis) that appear in both corpora and that present RF>1.

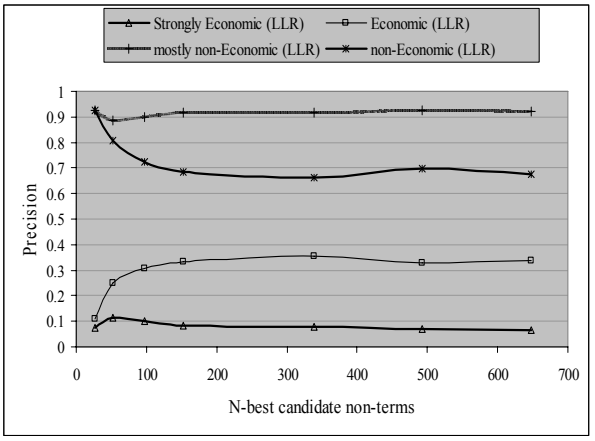


Figure 4. Precision (y-axis) for the N-best terms (x-axis) that appear in both corpora and that present RF<1.

Figure 5 depicts comparative results between LLR and term extraction based on simple frequency counts on DELOS only. This experiment was performed to show the importance of corpora comparison for term extraction, compared to using only a domain-specific corpus and applying simple frequencies to the candidate terms appearing in it. As expected, corpus comparison (LLR) leads to better results as it is concluded by the increased distance between the Economic term curves and the non-Economic term ones. Simple frequency counts tend to include many undesired N-grams among the candidate terms with the highest ranks, simply because these N-grams appear frequently in the corpus. As a result, the precision values with frequencies on one corpus only, inevitably drop.

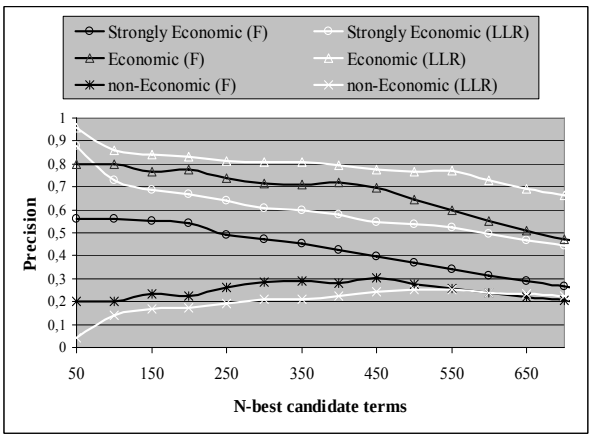


Figure 5. Comparative precision between LLR and simple frequency counts on DELOS.

Table 2 shows the RF and LLR scores of the 20 most highly ranked economic terms, ordered by their LLR value. The depicted counts are the original ones, prior to smoothing. An interesting term is "υψηλός", the ancient Greek form for "high", used today almost exclusively in the context of the degree of performance, growth, rise, profit, cost, drop (i.e. the appropriate form in economic context), as opposed to its modern form "ψηλός", which is used in the concept of the degree of actual height.

Table 2. The 20 most highly ranked economic terms

Rank	word	translation	Cw(D)	Cw(B)	RFw	LLR
1	εταιρία	company	5396	0	1845,9	852,0
2	δρχ	drachma	3003	1	342,5	465,5
3	μετοχή	stock	2827	6	74,4	414,0
4	αγορά	buy	2330	33	11,9	257,2
5	αύξησ η	growth, rise	2746	66	7,1	247,6
6	κέρδος	profit	1820	15	20,1	228,2
7	τράπεζα	bank	1367	11	20,3	171,8
8	επιχείρηση	enterprise	1969	56	6,0	162,1
9	κεφάλαιο	capital	1325	14	15,6	157,3
10	σημαντικός	important	1872	56	5,7	149,3
11	πώληση	sell	1203	11	17,9	147,3
12	προϊόν	product	1282	16	13,3	146,0
13	όμιλος	(company) group	1036	5	32,2	140,0
14	Α.Ε.	INC	820	0	280,7	126,4
15	μετοχικός	stocking	790	2	54,1	112,8
16	τιμή	price	1722	70	4,2	110,9
17	επιτόκιο	interest (financ.)	821	4	31,2	110,0
18	υψηλός	high (old form)	711	0	243,4	109,2
19	κόστος	cost	1031	19	9,0	103,4
20	κλάδος	branch	833	7	19,0	103,2

Figure 6 shows the difference in precision with LLR for the N-best terms with and without the application of smoothing. When smoothing is not applied, the drop in performance is significant (around 20%). The expected performance improvement due to the smoothing process is further enhanced, because the terms that appear only in DELOS (and not in the balanced corpus) are not taken into account when smoothing is not performed.

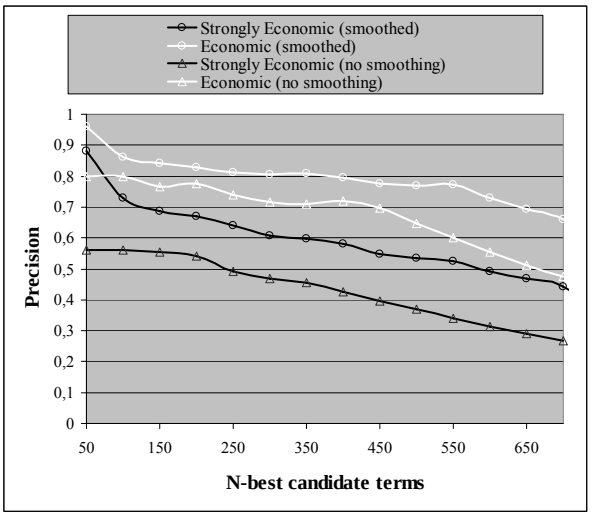


Figure 6. Comparative precision using the LLR metric with and without smoothing.

5 CONCLUSION

In this paper we have presented the process of automatically extracting economic terminology from Modern Greek texts. The properties of the language are taken into account by utilizing appropriate pre-processing tools. The linguistic complexity of the domain-specific corpus is addressed by adjusting the traditional candidate term formation methodology to deal with the freedom in word ordering. Finally, the unusual size difference between the two corpora (domain-specific and general) leads to a sparse data problem, which is dealt with satisfactorily by applying Lidstone's smoothing law.

ACKNOWLEDGEMENTS

We thank the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program PYTHAGORAS II, for funding the above work.

REFERENCES

[1] P. Drouin, '*Detection of Domain Specific Terminology Using Corpora Comparison*', 4th International Conference on Language Resources and Evaluation (LREC), 79–82, Lisbon, (2004).

[2] K.Frantzi, S. Ananiadou, and H. Mima, '*Automatic Recognition of Multi-word Terms: the C-value/NC-value Method*', International Journal on Digital Libraries, **3** (2), 117–132, (2000).

[3] N. Hatzigeorgiu, M. Gavrilidou, S. Piperidis, G. Carayannis, A. Papakostopoulou, A. Spiliotopoulou, A. Vacalopoulou, P. Labropoulou, E. Mantzari, H. Papageorgiou, and I. Demiros, '*Design and Implementation of the online ILSP Greek Corpus*', 2nd International Conference on Language Resources and Evaluation (LREC), Athens, 1737–1742, (2000).

[4] A. Hulth, '*Improved Automatic Keyword Extraction Given More Linguistic Knowledge'*, International Conference on Empirical Methods in Natural Language Processing (EMNLP), Sapporo, 216-223, (2003).

[5] K. Kermanidis, N. Fakotakis and G. Kokkinakis, '*DELOS: An Automatically Tagged Economic Corpus for Modern Greek*', 3rd International Conference on Language Resources and Evaluation (LREC), Las Palmas de Gran Canaria, 93-100, (2002).

[6] Kilgarriff, '*Comparing Corpora*', International Journal of Corpus Linguistics, **6** (1), 1-37, (2001).

[7] C. Manning and H. Schuetze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[8] R. Navigli and P. Velardi, '*Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites*', Computational Linguistics, **30** (2), 151–179, (2004).

[9] Partners of ESPRIT-291/860, *Unification of the Word Classes of the ESPRIT Project 860*, Internal Report BU-WKL-0376, (1986).

[10] K. Sgarbas, N. Fakotakis and G. Kokkinakis, '*A Straightforward Approach to Morphological Analysis and Synthesis*', Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries (COMLEX), Kato Achaia, Greece, 31–34, (2000).

[11] E. Stamatatos, N. Fakotakis and G. Kokkinakis, '*A practical chunker for unrestricted text*', Proceedings of the Conference on Natural Language Processing (NLP), Patras, Greece, 139–150, (2000).