Learning-free Text Categorization

Patrick Ruch, Robert Baud and Antoine Geissbühler

University Hospital of Geneva, Medical Informatics Division,
1205 Geneva, Switzerland
patrick.ruch@dim.hcuge.ch

Abstract. In this paper, we report on the fusion of simple retrieval strategies with thesaural resources in order to perform large-scale text categorization tasks. Unlike most related systems, which rely on training data in order to infer text-to-concept relationships, our approach can be applied with any controlled vocabulary and does not use any training data. The first classification module uses a traditional vector-space retrieval engine, which has been fine-tuned for the task, while the second classifier is based on regular variations of the concept list. For evaluation purposes, the system uses a sample of MedLine and the Medical Subject Headings (MeSH) terminology as collection of concepts. Preliminary results show that performances of the hybrid system are significantly improved as compared to each single system. For top returned concepts, the system reaches performances comparable to machine learning systems, while genericity and scalability issues are clearly in favor of the learning-free approach. We draw conclusion on the importance of hybrids strategies combining data-poor classifiers and knowledge-based terminological resources for general text mapping tasks.

1 Introduction

Typical concept mapping applications use a set of key-words as concepts to be selected into a *glossary*. However, key-word assignment is only the most popular application of such systems, and the task can also be seen as a named-entity (NE) recognition task if we consider entities that can be listed[1]. Computer-based concept mapping technologies include:

- *retrieval based on word-matching*, which attributes concepts to text based on shared words between the text and the concepts;
- *empirical learning of text-concept associations* from a training set of texts and their associated concepts.

Retrieval is often presented as the weakest method [2], however there are several areas of applications where training data are clearly missing [2].

[1] As it is the case in molecular biology with gene, protein and tissue entities [1]

[2] We must note that even if we would assume that large and representative training data will be once available for any possible domain, current machine learning sys-

1.1 Biomedical domain

To our knowledge the largest class set ever used by text classification systems is about 2.10^4, and such systems were applied to the biomedical domain, based on the Medical Subject Heading (MeSH) ontology. Although such a class set is already large for typical categorization tasks, terminological resources in health sciences, as well as documentation's needs require tools likely to process much larger sets of concepts[3].

1.2 Concept mapping as a learning-free classification task

General text classification has been largely studied and has led to an impressive amount of papers (see [4] for a recent survey of the domain). A non exhaustive list of machines learning approaches to text categorization includes naive Bayes[5]), k-nearest neighbors[4], SVM[6], boosting[7], and rule-learning algorithms[8]. However, most of these studies apply text classification to a small set of classes (usually a few hundreds, as in the paradigmatic Reuters' collection [9]). In comparison, our system is designed to handle large class sets: retrieval systems can be applied on a virtually infinite set of concepts and 10^{5-6} is still a modest range. For sake of evaluation the class set ranges from about 20,000 -if only unique canonical MeSH terms are taken into account- up to 140 000 -if synonyms are considered in addition to their canonical class.

1.3 MeSH mapping and MedLine

Figure 1 provides an example of a citation[4] in MedLine (authors, title, institution, and publication types are omitted; major MeSH terms are indicated with *; subheadings are removed and the semi-colon is used as separator) and its corresponding MeSH terms. Most text categorization studies working with MedLine collections neglect two important aspects of the MedLine's annotation with MeSH terms that will be considered in the present study:

a. availability of thesauri: the MeSH is provided with an important thesaurus (120,020 synonyms), whose impact will be assessed in our study;

---

tems still would have to face major scalability problems. The problem is twofold: it concerns both the ability of the system to work with large concept sets, and its ability to learn and generalize regularities for rare events: Larkey and Croft [3] show how the frequency of the concept in the collection is a major parameter for learning-base text categorization tools.

[3] Thus, the May 2002 release of the Unified Medical Language System (UMLS-2002AB) contained 871,584 different concepts and 2.1 million terms. In molecular biology, the SWISS-PROT Release 40.28 (September 2002) has 114033 entries, and most entries have synonyms, while the TrEMBL Release 21.12 (September 2002) has 684666 entries.

[4] It must be observed that MedLine's annotation is done on the basis of the complete article, while in our experiments only the abstract is considered.

b. comprehensiveness: the MeSH follows a hierarchical structure, but if we consider only unique strings, there are 19 632 terms; unlike related results (discussed in section 4.1), our system is applied with the full MeSH.

The production of exopolysaccharides (EPSs) by a mucoid clinical isolate of Burkholderia cepacia involved in infections in cystic fibrosis patients, was studied. Depending on the growth conditions, this strain was able to produce two different EPS, namely PS-I and PS-II, either alone or together. PS-I is composed of equimolar amounts of glucose and galactose with pyruvate as substituent, and was produced on all media tested. PS-II is constituted of rhamnose, mannose, galactose, glucose and glucuronic acid in the ratio 1:1:3:1:1, with acetate as substituent, and was produced on either complex or minimal media with high-salt concentrations (0.3 or 0.5 M NaCl). Although this behavior is strain-specific, and not cepacia-specific, the stimulation of production of PS-II in conditions that mimic those encountered by B. cepacia in the respiratory track of cystic fibrosis patients, suggests a putative role of this EPS in a pathologic context.

Burkholderia cepacia*; Carbohydrate Conformation; Carbohydrate Sequence; Comparative Study; Culture Media*; Cystic Fibrosis*; Glucose; Glycerol; Human; Molecular Sequence Data; Onions; Phenotype; Polysaccharides, Bacterial*; Temperature

Fig. 1. Citation of MedLine with MeSH terms provided by professional indexers.

The remainder of this paper is organized as follows: the next section presents the collection and metrics used, as well as the basic classifiers. Then, we describe and evaluate our basic classifiers, before presenting and testing how these classifiers can be merged. The performance of the combined mapping system is compared to related studies. Finally, we conclude and suggest some future work.

2 Evaluation

Following [3] and as it is usual with retrieval systems, the core measure for the evaluation is based on the 11-point average precision. We provide the total number of relevant terms returned by the system on the complete run. The top precision (interpolated $Precision_{at\ Recall=0}$) is also given. In order to provide a minimal assesment of the system, we apply the system on the Cystic Fibrosis[5] (CF) collection [10], The CF collection is a collection of 1239 MedLine citations. For each citation, we used the content of the abstract field as input in the system. Using other fields, such as the title or the publication's source may have provided interesting additional evidences for classification, but we decided to work only with the abstract in order to minimize the number of variables to be controlled. The average number of concepts per abstract in the collection is 12.3 and the following measures were done considering the top-15 terms returned (TR).

[5] Available on Marti Hearst's pages at http://www.sims.berkeley.edu/ hearst/irbook/

3 Method

One of the most comprehensive study of MeSH classification based on simple word-matching has been carried at the National Library of Medicine and has led to the development of the MetaMap tool. For developing MetaMap, different methods and combination of methods were compared [11], including retrieval strategies (based on INQUERY distance metrics), syntactic and statistical phrase chunking, and MeSH coocurrences. Unfortunately the system has been evaluated on the UMLS collection, which is not publicly available. We use the UMLS distribution of the MetaMap system with the MeSH as concept list and with default settings in order to obtain a blackbox baseline measure for comparison with our systems. Table 1 shows the results of MetaMap, together with the two basic classifiers, which are going to be described in the next section. We see that MetaMap outperforms each classifier on the complete Cystic Fibrosis collection.

3.1 Basic classifiers

Table 1. Results for MetaMap, RegEx, and (tf.idf) classifiers. weighting schemas. For the VS engine, tf.idf parameters are provided: the first triplet indicates the weighting applied to the "document collection", i.e. the concepts, while the second is for the "query collection", i.e. the abstracts. The total of relevant terms is 15193.

System or parameters	Relevant retrieved	Top precision	11pt Average precision
MetaMap	4075	.7425	.1790
RegEx	3986	.7128	.1601
tf.idf (VS)			
lnc.atn	3838	.7733	.1421
anc.atn	3813	.7733	.1418
ltc.atn	3788	.7198	.1341
ltc.lnn	2946	.7074	.111

Two main modules constitute the skeleton of our system: the regular expression component, and the vector space component. The former component uses tokens as indexing units and can take advantage of the thesaurus, while the latter uses stems (Porter-like). Each of the basic classifiers uses known approaches to document retrieval. The first tool is based on a regular expression pattern matcher. Although such approach is less used in modern information retrieval systems [6], it is expected to perform well when applied on very short documents such as key-words: MeSH terms do not contains more than 5 words. The second classifier is based on a SMART-like vector-space engine[13]. This second tool

[6] With a notable exception, the GLIMPSE system [12].

is expected to provide high recall in contrast with the regular expression-based tool, which should privilege precision.

Regular expressions and MeSH thesaurus The regular expression (RegEx) pattern matcher is applied on the the canonic list of MeSH terms (19 936 concepts) augmented with its thesaurus (the total includes 139 956 terms). In this system, text normalization is mainly processed by removing punctuation or by the MeSH terminological resources when the thesaurus is used. Indeed, the MeSH thesaurus provides a large set of "synonyms", which are mapped to a unique MeSH representative in the canonic collection. Instead of synonyms, this set gathers morpho-syntactic variants (mainly for plural forms), noun phrase reformulations, strict synonyms, and a last class of related terms, which mixes up generic terms, specific terms, and some other kinds of less obvious semantic relations: for example. *Inhibition* is mapped to *Inhibition (Psychology)*. The manually crafted transition network of the pattern-matcher is very simple, as it allows some insertions or deletions within a MeSH term, and ranks the proposed candidate terms based on these basic edit operations following a completion principle: the more terms are matched, the more the term is relevant. The system hashes the abstract into 5 token phrases and moves the window through the abstract. The same type of operations is allowed at the token level, so that the system is able to handle minor string variations, as for instance between *diarrhea* and *diarrhoea*. Unexpectedly, table 1 shows that the single RegEx system performs better than any single tf.idf[7] (term frequency-inverse document frequency) system, so that surprisingly, the thesaurus-powered pattern-matcher provides better results than the basic VS engine for MeSH mapping.

Vector space system The vector space (VS) module is based on a general IR engine[8] with *tf.idf* weighting schema. In this study, it uses stems (Porter-like, with minor modifications) as indexing terms, and a stop word list. While stemming can be an important parameter, whose impact is sometimes a matter of discussion [15], we did not notice any significant differences between the use of tokens and the use of stems, while the index's size is larger (8755 vs. 5972 entries) when tokens are chosen as indexing units. The graceful behavior of stemming is probably due to the fact that tokens of the biomedical vocabulary are usually longer that in regular English, so that word conflation creates only few confusing stems. However, we noticed that a significant set of semantically related stems should have been conflated in the same indexing unit: for example, the morpheme *immun* is found in 48 different stems, and using a morpheme-based word conflation system could have improved the system. Finally, let us note that MeSH terms contain 1 to 5 words, so that, we could have used phrases

[7] We use the (de facto) SMART standard representation in order to express these different parameters, cf. [14] for a detailed presentation. For each triplet provided in table 1, the first letter refers to the *term frequency*, the second refers to the *inverse document frequency* and the third letter refers to a *normalization factor*.

[8] Available on the first author's homepage.

(as in [16] and [17]), however, we believe that part of the improvement that could have been brought by using phrases is probably achieved by the RegEx module.

A large part of this study was dedicated to tuning the VS engine, and tf.idf weighting parameters were systematically evaluated. The conclusion is that cosine normalization was especially effective for our task. This is not surprising, considering the fact that cosine normalization performs well when all documents are short as is the case of MeSH terms [9]. Thus, in table 1, the top-4 weighting function uses cosine as normalization factor. We also observed that the *idf* factor, which was calculated on the MeSH collection performed well, it means that the canonical MeSH vocabulary is large enough to effectively underweight non-content words (such as *disease* and *syndrome*). Calculating the idf factor on a collection of a large collection of abstracts could have been investigated, but such solution may have resulted in making the system more collection-dependent.

4 Results

The hybrid system combines the regular expression classifier with the vector-space classifier. Unlike [3] we do not merge our classifiers by linear combination, because the RegEx module does not return a scoring consistent with the vector space system. Therefore the combination does not use the RegEx's score, and instead it uses the list returned by the vector space module as a *reference* list (RL), while the list returned by the regular expression module is used as *boosting* list (BL), which serves in order to improve the ranking of terms listed in RL. A third factor takes into account the length of terms: both the character's length (L_1) and the byte size (L_2, with $L_2 > 3$) of terms are computed, so that long and/or multi-word terms appearing in both lists are favored over short and/or single word terms. We assume that the reference list has exhaustive coverage, and we do not set any threshold on it. For each term t listed in the RL, the combined Retrieval Status Value (RSV) is:

$$RSV_{Hybrid} = \begin{cases} RSV_{VS}(t) \cdot Ln(L_1(t) \cdot L_2(t) \cdot k) & \text{if } t \in BL, \\ RSV_{VS}(t) & \text{otherwise.} \end{cases} \quad (1)$$

Table 2 shows that the optimal tf.idf parameters *lnc.atn* for the basic VS classifier does not provide the optimal combination with RegEx. The optimal combination is obtained with *ltc.lnn* settings [10]. We also observe that the *atn.ntn* weighting schema maximizes the top candidate (i.e. $Precision_{at\ Recall=0}$) measure, but for a general purpose system, we prefer to maximize average precision, since this is the only measure that summarizes the performance of the full ordering of concepts. However, in the context of a fully automatic system (for example for CLIR purposes), the top-ranked concepts (1 or 2) are clearly of major importance, therefore we also provide this measure.

[9] As for more advanced schema, we tested the combination of RegEx with pivoted normalization and it did not outperform the combination RegEx + ltc.lnn.

[10] For the augmented term frequency factor (noted a, which is defined by the function $\alpha + \beta \times (tf/max(tf))$, the value of the parameters is $\alpha = \beta = 0.5$.

Table 2. Combining VS with RegEx

Weighting function concepts.abstracts	Relevant retrieved	Top Precision	Average Precision
Hybrids: tf.idf (VS) + RegEx			
ltc.lnn	4308	.8884	.1818
lnc.lnn	4301	.8784	.1813
anc.ntn	4184	.8746	.1806
anc.ntn	4184	.8669	.1795
atn.ntn	3763	.9143	.1794

4.1 Related results

While several works have concentrated on applying machine learning methods to text categorization, it is often difficult to compare and synthesize the wide quantity of results provided in these studies. One of the main reason is probably that there is no strict definition of the task, which we believe must be seen as a subtask [11] rather than a task in itself. Indeed, apart from the central classification problem and the common textual material, which are shared by all these subtasks, there are few common points between them. The gap is well exemplified if we consider on the one side TC applied to sentence extraction, like it is usual in automatic summarization, and on the other side concept mapping: while the former work with a couple of classes (up to a dozen in [18] or [19]), the latter uses virtually infinite sets of classes. Between the two edges, a continuous span of text classification experiments can be identified, whose the most studied -which can also be seen as the paradigmatic ones- are centrally located from some hundreds up to some thousands of classes.

OHSUMED As for classification with the MeSH and using MedLine records, the OHSUMED collection has been often used. Like the CF collection, the OHSUMED collection contains a list of MedLine abstracts, so that both collection are equally representative of MedLine. To the best of our knowledge only two studies have used the entire set of 14,000 MeSH categories [20] [21] used in OHSUMED, an no one ever used the complete 20000-items MeSH terminology that we used, therefore comparison is difficult. The main reason for this is that many TC methods cannot process such large sets. Yang [20], Lewis et al. [22], and Lam and Ho [21] have published results using the subset of categories from the "Heart Diseases" sub-tree (HD-119, so-called because it uses only 119 concepts). In [23], 42 categories of the HD sub-tree were excluded, because these categories had a training set frequency less than 15. Yang [20] reduces the collection to only those documents that are positive examples of the categories of the HD-119. The final profile of the test collection is very different as the number of terms per abstract is 1.4. Joachims [6] has also published results for the OHSUMED collection using SVM. His work uses the first 20,000 documents of the year 1991

[11] However, document filtering as in TREC-9 is a real task.

divided into two subsets of 10,000 documents each that are respectively used for training and testing. He reports on very impressive results but his TC task is very different: he assumes that if a category in the MeSH tree is assigned then its more general category in the hierarchy is also present, so that he uses only the high level disease categories. This simplifies the task considerably and may partially explain the good results obtained in these experiments.

Nevertheless, we still attempt to provide some elements for comparing our system with previous studies. The most similar experiment was probably conducted by [24] (noted YC in the following). The authors use a classifier based on singular value decomposition (LLSF) for text categorization. They use the international Classification of Diseases (ICD) as concept list, and full-text diagnosis as instances to be classified. ICD -like the MeSH- contains a large number of categories (about 12 000), and is also provided with an important thesaurus. Both collections are lexically related: we can notice that most of the 6000 diseases listed in the MeSH subtree for diseases have an equivalent in ICD codes, so that ICD can be seen as a more specific partition of the MeSH categories restricted to the "disease" subtree. So assigning ICD codes and MeSH terms are quite similar tasks and supports a possible comparison. Unfortunately only comparison with $Precision_{atRecall=0}$ is available in their study. We also indicates the results obtained by the SMART system as reported in [25] (noted YY in the following). Even if she works with about 4000 MeSH terms only, this result is useful in order to provide a common baseline measure.

Method/Collection/Paper	Av. Prec.	$Prec_{atRec=0}$
SMART/OHSU/YY	.15 (0)	.61 (0)
LLSF/ICD	-	.840 (+37.7)
ltc.lnn/CFC	.1818 (+20.0)	.8884 (+45.0)
atn.ntn/CFC	-	.9143 (+49.9)

Table 3. Comparison: our hybrid system vs. learning systems. Weighting schema are given for the VS system.

Comparison measures are reported in table 3. For top precision, we observe that our hybrid system (+45.9 for *ltc.lnn* and +50.9 for *atn.ntn*) is more efficient than LLSF (+37.7%). Now, regarding average precision, our method outperforms SMART by 25%.

Finally, these results are opposite to what is concluded in [20]: simple word-based strategies behaves gracefully as concept granularity grows, i.e. the more concepts there are in a collection, the more effective retrieval strategies will be. We can assume that retrieval approaches perform well when categories are numerous, not only because training becomes a major issue for learning systems

[12], but because the high granularity may help the retrieval system to cover every dimension of the conceptual space. On the opposite, learning systems are able to infer and cluster categories (generic or specific concepts) that are not explicitly present in the source document, so high granularity does not really help them.

5 Conclusion and future work

Concept mapping can be seen both as an alternative to scalability issues of learning methods and as a complementary module -as IR systems are often- likely to provide solution, when training data are insufficient. In a medium position, concept mapping can be seen as an optional module, as it provides a strategy to classify along these classes that are absent or subrepresented in the training data.

We have reported on the development and evaluation of a MeSH mapping tool. The systems combines a pattern matcher, based on regular expressions of MeSH terms, and a vector space retrieval engines that uses stems as indexing terms, a traditional *tf.idf* weighting schema, and cosine as normalization factor. The hybrid system showed results similar or better to machine learning tools for the top returned candidate terms, while scalability of our data-poor (if not -independent) approach is also an advantage as compared to data-driven system. The system provides a new baseline for text categorization systems, improving average precision by 20% in comparison to standard retrieval engines (SMART). Finally, combining learning and learning-free systems could be beneficial in order to design general broad-coverage concept mapping systems.

Acknowledgements

The study has been partially sponsored by the European Union (IST Grant 2001-33260, see www.wrapin.org) and the Swiss National Foundation (Grant 3200-065228).

References

1. Shatkay, H., Edwards, S., Wilbur, W., Boguski, M.: Genes, themes and microarrays: using information retrieval for large-scale gene analysis. Proc Int Conf Intell Syst Mol Biol **8** (2000) 317–28
2. Yang, Y.: Sampling strategies and learning efficiency in text categorization. In: Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access. (1996)
3. Larkey, L., Croft, W.: Combining classifiers in text categorization. In: SIGIR, ACM Press, New York, US (1996) 289–297
4. Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval **1** (1999) 67–88

[12] Again, this problem is avoided in studies conducted with learning systems by filtering out concepts with low frequencies.

5. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification (1998)
6. Joachims, T.: Making large-scale svm learning practical. Advances in Kernel Methods - Support Vector Learning (1999)
7. Schapire, R., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning **39** (2000) 135–168
8. Apté, C., Damerau, F., Weiss, S.: Automated learning of decision rules for text categorization. ACM Transactions on Information Systems (TOIS) **12** (1994) 233–251
9. Hayes, P., Weinstein, S.: A system for content-based indexing of a database of news stories. Proceedings of the Second Annual Conference on Innovative Applications of Intelligence (1990)
10. Shaw, W., Wood, J., Wood, R., Tibbo, H.: The cystic fibrosis database: Content and research opportunities. LSIR **13** (1991) 347–366
11. Aronson, A., Bodenreider, O., Chang, H., Humphrey, S., Mork, J., Nelson, S., Rindflesch, T., Wilbur, W.: The indexing initiative. A report to the board of scientific counselors of the lister hill national center for biomedical communications. Technical report, NLM (1999)
12. Manber, U., Wu, S.: GLIMPSE: A tool to search through entire file systems. In: Proceedings of the USENIX Winter 1994 Technical Conference, San Fransisco CA USA (1994) 23–32
13. Ruch, P.: Using contextual spelling correction to improve retrieval effectiveness in degraded text collections. COLING 2002 (2002)
14. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. ACM-SIGIR (1996) 21–29
15. Hull, D.: Stemming algorithms: A case study for detailed evaluation. Journal of the American Society of Information Science **47** (1996) 70–84
16. Tan, C., Wang, Y., Lee, C.: The use of bigrams to enhance text categorization. Information Processing and Management **38** (2002) 529–546
17. Aronson, A.: Exploiting a large thesaurus for information retrieval. Proceedings of RIAO (1994) 197–216
18. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Research and Development in Information Retrieval. (1995) 68–73
19. McKeown, K., Barzilay, R., Evans, D., Hatzivassiloglou, V., Schiffman, B., Teufel, S.: Columbia multi-document summarization: Approach and evaluation. In: Proceedings of the Workshop on Text Summarization, ACM SIGIR Conference 2001, (DARPA/NIST, Document Understanding Conference)
20. Yang, Y.: An evaluation of statistical approaches to medline indexing. AMIA (1996) 358–362
21. Lam, W., Ho, C.: Using a generalized instance set for automatic text categorization. In: SIGIR. (1998) 81–89
22. Lewis, D.: Evaluating and Optimizing Autonomous Text Classification Systems. In: SIGIR, ACM Press (1995) 246–254
23. Lewis, D., Shapire, R., Callan, J., Papka, R.: Training algorithms for linear text classifiers. In: SIGIR. (1996) 298–303
24. Yang, Y., Chute, C.: A linear least squares fit mapping method for information retrieval from natural language texts. COLING (1992) 447–453
25. Yang, Y.: Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. In Croft, W., van Rijsbergen, C., eds.: SIGIR, ACM/Springer (1994) 13–22