Retrieving hierarchical syllabus items for exam question analysis

John Foley & James Allan

Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
{jfoley, allan}@cs.umass.edu

Abstract. Educators, institutions, and certification agencies often want to know if students are being evaluated appropriately and completely with regard to a standard. To help educators understand if examinations are well-balanced or topically correct, we explore the challenge of classifying exam questions into a concept hierarchy.

While the general problems of text-classification and retrieval are quite commonly studied, our domain is particularly unusual because the concept hierarchy is expert-built but without actually having the benefit of being a well-used knowledge-base.

We propose a variety of approaches to this "small-scale" Information Retrieval challenge. We use an external corpus of Q&A data for expansion of concepts, and propose a model of using the hierarchy information effectively in conjunction with existing retrieval models. This new approach is more effective than typical unsupervised approaches and more robust to limited training data than commonly used text-classification or machine learning methods.

In keeping with the goal of providing a service to educators for better understanding their exams, we also explore interactive methods, focusing on low-cost relevance feedback signals within the concept hierarchy to provide further gains in accuracy.

1 Introduction

Educators use exams to evaluate their students' understanding of material, to measure whether teaching methodologies help or hurt, or to be able to compare students across different programs. While there are many issues with exams and evaluations that could be and are being explored, we are interested in the question of coverage – whether an evaluation is complete, in the sense that it covers all the aspects or concepts that the designer of the evaluation hoped to cover.

We consider classifying multiple-choice questions into a known concept hierarchy. In our use case, an educator would upload or enter an exam into our system, and each question would be assigned to a category from the hierarchy. The results would allow the educator to understand and even visualize how the

questions that make up the exam cover the overall hierarchy, making it possible to determine if this coverage achieves their goals for the examination: are all important topics covered?

This problem is traditionally treated as one of manual question creation and labeling, where an official, curated set of tests has been created and are to be used widely or repeatedly. Educators who use that exam are guaranteed "appropriate" coverage of the material. However, this centralized approach is only a partial solution to the problem of understanding coverage of exams since every institution and almost every teacher or professor is likely to have their own assignments, their own quizzes, their own exams. The global exam does not help those educators understand how their own material fits into the known set of topics.

For this study, our dataset is a medium-sized corpus of test questions classified into the American Chemical Society (ACS) hierarchy developed by their exams institute [12]. This dataset has been used for educational research [11, 17], but as these are actual exams used by educators, it is not available publicly.

The problem is interesting because the hierarchy is crisply but very sparsely described and the questions are very short, on par with the size of microblog entries. In existing text classification datasets with a hierarchical components (e.g., Wikipedia categories, the Enron email folder dataset [14], and the Yahoo! Directory or Open Directory Project [26]) all of the labeled documents are quite dense, the categories were created with various levels of control, and the resulting categories are likely to be overlapping. In contrast, all of our information is sparse, the categories themselves were designed by experts in the field, and part of their goal was to have questions fall into a single category.

In this study, we explore methods for classifying exam questions into a concept hierarchy using information retrieval methods. We show that the best technique leverages both document expansion and concept-aware ranking methods, but that exploiting the structure of the questions is helpful but not shown to be an advantage in conjunction with our other approaches on this dataset.

Ideally this work would be repeated on additional sets of questions with their own hierarchy to show its broad applicability; unfortunately, such questions are carefully guarded [1] and difficult to come by so demonstrating the results on another dataset must be left for future work.

Although our evaluation dataset is not open, we believe the results will apply to any comparable collection of exam questions categorized into a known hierarchy and we hope that our success in this task will encourage other educators and institutions to open up their data and new problems to our community. Our key approach leverages structure present in this kind of dataset that is not available in standard retrieval collections, but we hope to explore its generality in future work.

[1] Even most standardized tests require test-takers to sign agreements not to distribute or mention the questions, even after the exam is taken.

2 Related Work

The problem we tackle in this study is classification of short text passages into a hierarchical concept hierarchy, sometimes with interaction. The classification of short texts is relevent even though we do not have sufficiently balanced training labels for our task. Additional prior work involves interactive techniques as well as hierarchical retrieval models.

Our domain is exam questions in chemistry. We have found very little existing work within this domain of education-motivated IR. Omar et al. [20] develop a rule-based system for classifying questions into a taxonomy of learning objectives (do students have knowledge, do they comprehend, etc.) rather than topics. They work with a small set of computer programming exam questions to develop the rules but do not actually evaluate their utility for any task.

The problem of question classification [30, 18] seems related but refers to categorizing informational questions into major categories such as who, where, what, or when.

2.1 Short-Text Classification

There is a huge body of literature on the well known problem of text classification, with a substantial amount devoted to classifying short passages of text. We sketch the approaches of a sample of that work to give an idea of the major approaches. Rather than attempt to cover it here, we refer the interested reader to the survey by Aggarwal and Zhai [1].

Sun et al. [26] considers a problem similar to ours, classifying short web page descriptions into the Open Directory Project's hierarchy. In their work, classification is done in two steps: the 15 categories most similar to the text are selected from the larger set of over 100,000 categories, and then an SVM is used to build a classifier for just those 15 categories so that the text can be categorized. Their category descriptions are selected by tf·idf comparison as well as using "explicit semantic analysis" [8]. Following related earlier work by Xue et al. [29], they represent an inner node of the hierarchy by its own content as well as that of its descendants. We represent leaf nodes by the content of their ancestors as well as their descendants, and try this in conjunction with document expansion.

Ren et al. [23] consider the problem of classifying a stream of tweets into an overlapping concept hierarchy. They treat the problem as classification rather than ranking, and do not explore interactive possibilities. They expand the short texts using embedded links and references to named entities and address topic drive using time-aware topic modeling, approaches that have little utility when processing exam questions. Banerjee et al. [3] effectively expand text by retrieving articles from Wikipedia and using the titles of those articles as features. By contrast, we expand text using an unlabeled set of questions – that is, comparable instances of the items we are classifying, having found wikipedia to not be helpful in such a focused domain.

A similar result with information retrieval applications comes from Dumais and Chen [6], who consider the problem of classifying search engine snippets into

a hierarchy with the goal of presenting an organization of the pages. They used SVM as a classifier, but worked only with the top levels of the category that had numerous training instances, unlike in our case where we have no training data.

While we have similarities to prior work in this space, we must reiterate that we used these works as inspiration and that bringing them to an *unsupervised setting* and validating the approaches in a new domain is a contribution.

2.2 Hierarchical Retrieval Models

The hierarchical retrieval models we propose and evaluate in this work draw inspiration from hierarchical classification. They also share some similarities with cluster-based retrieval [16], in the way that a document is represented by its terms and those of its cluster, we will represent nodes based on their features and the features belonging to their parents. Hierarchical language models show up in the task of expert finding as well, given the hierarchy of employees in the company [21, 2]. Our task differs from expert retrieval in that the elements of our hierarchy are precisely defined by their own descriptions, but do not interact with documents in any way.

Lee et al. present an early work on leveraging a hierarchy in the form of a knowledge-base graph, constructed mostly of "is-a" relationships [15]. Ganesan et al. present a work on exploiting hierarchical relationships between terms or objects to compute similarity between objects that are expressed in terms of elements in the hierarchy [9], while relevant, this would be of more use if we were trying to match exams to other exams.

2.3 Interactive Learning

Active learning [24] is an approach to classification that allows the learning algorithm to select some instances of data for labeling, with the idea that some subset of labels is better for training than all of those available. Although this does reduce labeling effort, it is not typically directed at reducing user labels for a task.

Hoi et al. [10] explored batch active learning approaches for classification of web pages and news articles, all of which are much longer than the exam questions we consider. They explore the learning curves for 10s or 100s of labels rather than the single interaction we consider (we can't expect 10s of labels per question a user wishes to classify, but one is more reasonable).

Bekkerman et al. [4] showed that a classifier could be improved by allowing a user to correct or augment the word features that were selected. If we consider the high-level concepts as added features, our approach is related, though they focus on document clustering rather than classification and use quite different collections.

3 Nodes, Questions, and Exams

The dataset we explore in this work is a collection of Chemistry exams created by the American Chemical Society (ACS) and a hierarchical taxonomy for those questions. We make the claim that these exams, in conjunction with the nodes in the hierarchy, are an interesting and challenging dataset. Although this dataset has been used in other studies [12, 11, 17], this paper must introduce it to our field. In this section, we discuss the format of the data, and some of our observations about its composition and distribution.

3.1 Concept hierarchy

Table 1. Top-level children in the ACS General Chemistry Hierarchy.

I	Atoms	**VI**	Energy and Thermodynamics
II	Bonding	**VII**	Kinetics
III	Structure and Function	**VIII**	Equilibrium
IV	Intermolecular Interactions	**IX**	Experiments
V	Chemical Reactions	**X**	Visualization

The concept hierarchy designed by the ACS has four levels, excluding the root "General Chemistry" node. Each level has a distinguishing numbering system. The top level of the hierarchy are identified as *Anchoring Concepts*, or Big Ideas. These are listed in Table 1.

Each of the nodes described in the hierarchy has a succinct description, but only the nodes in Table 1 have titles, i.e.

X. Visualization Chemistry constructs meaning interchangeably at the particulate and macroscopic levels.

X.A.2.a. Schematic drawings can depict key concepts at the particulate level such as mixtures vs. pure substance, compounds vs. elements, or dissociative processes.

The there are ten nodes at the first level, as already discussed, 61 at the level below that, 124 at the third level, and 258 leaf nodes. Of the middling nodes, there are between 1 and 10 children assigned to each, with most of the weight belonging to 1, 2 and 3 (72, 59, and 37 respectively). The average length of a node description is 18.3 terms, and there are 16.2 distinct terms per node.

3.2 Exam questions

An exam question looks like the following, except it is slightly too broad and lacks multiple choice solutions:

> I know sulfuric acid is an important catalyzer and is used in various processes. My question is, how do I recover the remaining sulfuric acid? It will be impure, and I don't know how to do the "standard" procedure (is there one?)[2]

The exam question has three parts. The *context* "sulfuric acid is an important catalyzer" presents the background for the question, giving the background details that are needed to know what the question means and how to pick an answer. The *question statement* itself "how do I recover the remaining sulfuric acid?" is the actual statement. In many cases, a single context will occur with several different questions, a factor that complicates simple comparison of the entire exam question. Finally, the exam question has the *answers*, usually multiple choice and usually with only know of them a correct answer. We did not find question fields to be helpful in the presence of our other, less-domain-specific ideas.

The ACS dataset includes 1593 total questions, distributed across 23 exams, with an average of 69 questions per exam. One exam has only 58 questions, and the largest exam has 80.

The most frequently tested concepts are tested tens of times over all these exams, the most frequent occurring 47 times – on average twice per exam for 23 exams. This most common node belongs to the "experimental" sub-tree, and discusses the importance of schematic drawings in relation to key concepts. It is one of the more general nodes we have inspected. The other most frequent concepts include "quantitative relationships and conversions," "moles," and "molarity".

The labeled data itself is highly skewed overall. There are 65 nodes that have ten or more questions labeled to belong to them. There are 62 nodes that only have a single question and another 29 that only have two questions – the number of rarely-tested nodes are the reason we choose to eschew supervised approaches in this work.

4 Evaluation Measures

Our task is ultimately to classify an exam question into the correct leaf node of the concept hierarchy. In part to support reasonable interactive assistance, we treat this as a ranking problem. That is, rather than identify a single category for a question, we generate a ranked list of them and evaluate where the correct category appears in the list.

An individual question's ranking is measured by two metrics. We use reciprocal rank (RR), the inverse of the rank at which the correct category is found. If there are multiple correct categories (uncommon), the first one encountered in the list determines RR. We also use normalized discounted cumulative gain (NDCG)

[2] `User Fiire; http://chemistry.stackexchange.com/questions/4250. This example displayed in lieu of the proprietary ACS data.`

as implemented in the Galago search engine[3] and formulated by Järvelin and Kekäläinen [13]. Additionally, we look at precision at rank 1, (P@1) because it represents the classification precision, if the rest of the ranking were to be ignored.

Since we are given exams as natural groupings of questions, and one of the key use-cases of our system will be the categorization of pre-existing exams for analysis, we evaluate our abilities on a per-exam level, rather than on a per-question level. That means that the accuracy for individual questions is averaged to create a per-exam average score. Formally, our reported scores are calculated as follows:

$$score = \frac{1}{|E|} \sum_{e \in E} \frac{1}{|Q_e|} \sum_{q \in Q_e} m(q)$$

where e is a single exam from E, the set of 23 exams, Q_E is the set of questions on exam e, and $m(q)$ is either RR, NDCG or P@1 for a query. This mean of averages is a macro-averaged score. We investigated whether micro-averaging (with each question treated equally rather than as part of an exam) made a difference, but there was no effect on the outcome of any experiment. As a result, we only report the score as described above.

5 Question-Framework Linking Methods

As mentioned previously, we consider our task to be one of retrieval, and not of classification, as we do not have training data for each of our labels. In this framework, each question is a query, and the corpus documents are the nodes or "labels" in the hierarchy (particularly the leaf nodes, but sometimes interior nodes). Therefore, we begin by using state-of-the-art retrieval models [19] and existing techniques like document expansion (section 5.1). Our best improvement comes from an extension to our retrieval model which incorporates parent/child relations in the concept hierarchy (section 5.3).

Our baseline is SDM, the sequential dependence model [19] which is known to be a highly effective ranking algorithm. Table 2 shows the results for the baseline in the top row. We also considered the query likelihood (QL) similarity [22], but SDM incorporates term dependencies in the context of bigrams and unordered window features. For all techniques, SDM was superior, so we do not report the unigram model (QL) numbers here. Our language model approach to retrieval is equivalent to a language-modeling approach to text-classification, but we present our ideas in the light of information retrieval for ease of implementation and evaluation.

5.1 Unsupervised Node Expansion

Both our corpus documents (concepts) and queries (questions) are short, so vocabulary mismatch – wherein a query and document are relevant but have

[3] `http://lemurproject.org/galago.php`

little or no words in common – is quite likely. One way we address that is to expand the concept descriptions with synonyms and strongly related words or phrases.

We use document expansion to accomplish that. To apply document expansion, we look for highly similar "neighbor" documents in an additional, external data source to help to improve the representation of the original documents for retrieval. It has been used for numerous purposes and has been explored thoroughly in prior work [27, 25, 7].

We use a publicly-available Q&A dataset[4] where all questions and comments are likely to be on or near the topic of chemistry, and used it as our expansion corpus. We briefly explored leveraging Wikipedia as in related work [28, 3, 8, 5], but initial experiments gave poor results: Wikipedia articles match too many nodes in our hierarchy; again, results are withheld for space.

For node expansion, we explored expansion with $k = \{1, 5, 10, 25, 50, 100\}$ Q&A comments or posts. We selected the neighbors using SDM, because it is known to perform well. Table 2 shows the substantial gain provided by node expansion (NX-50) before using SDM. We selected an expansion by 50 neighbors based on training data.

5.2 Question Context Model

Recall that the exam questions include three parts: the context, the statement, and the answers. We hypothesized that this structure could be leveraged to improve matching of exams. Indeed, the context can appear in multiple questions that are categorized differently, so although it is important, it also may be a distractor. We define the QCM similarity between two questions as:

$$QCM(q_i, q_{i+1}) = \lambda \text{SDM}_S(q_i, q_{i+1}) + (1 - \lambda)\text{SDM}_C(q_i, q_{i+1})$$

where q_i and q_{i+1} are two questions, SDM_S is the question statement similarity between them, and SDM_C is the similarity between the contexts.

5.3 A Hierarchy-Aware Retrieval Model

Drawing inspiration from hierarchical classification techniques, we propose a model of retrieval that takes into account the construction of the hierarchy, namely, that any node N in the hierarchy is described not just by its text, but also by the text of its ancestors and descendants. A low-level node about how to measure the density of a liquid is partially described by its highest level node, which encompasses all experimental techniques.

Hierarchical Node Scoring The score of a leaf node given a query is given by a retrieval model. As mentioned above, we use the SDM approach for these experiments. However, if a query matches a leaf node well but does not match

[4] The beta version of `chemistry.stackexchange.com`.

the parent of the leaf node, the match is suspect and should be down-weighted. To accommodate that, we use a hierarchical SDM scoring approach.

We first define an operator that returns the ancestors of a node, $A(N)$, excluding the root itself. This operator is defined inductively, using the operator $P(n)$ that returns the parent of node n.

$$A(N) = \begin{cases} \varnothing & N \text{ is a root node} \\ N \cup A(P(N)) & \text{otherwise} \end{cases}$$

We choose to exclude the root node because it has no description in our hierarchy.

Given the set of ancestors $A(N)$ of any node N, we can assign a joint score to the nodes based upon its score and that of its ancestors. If $\text{SDM}(q, N)$ is the SDM score for node N with query q, then:

$$\text{H-SDM}(q, N) = \prod_{n \in A(N)} \text{SDM}(q, n)$$

Descendant Node Expansion (NX) In addition to generating and combining scores for all nodes on a path to the root, we can accomplish a similar purpose by instead expanding nodes such that they are explicitly represented by the text of their descendants. We define an operator $D(N)$ which collects the set of descendants for a given node, given an operator $C(N)$ which returns a set of children the node N.

$$D(N) = \begin{cases} N & N \text{ is a leaf node} \\ N \cup \{D(c) \mid c \in C(N)\} & \text{otherwise} \end{cases}$$

We use this pattern to select nodes to expand the representation of the nodes in our model. This pattern leverages the intuition that experimental techniques (child IX of the root node) could be better represented by all of the experimental techniques available in the hierarchy in addition to its succinct description.

5.4 Experimental Results

Table 2. Evaluation of Methods

Model	MRR	NDCG	P@1
SDM	0.179	0.311	0.090
QCM	0.188	0.319	0.090
SDM (NX-50)	0.263	0.398	0.144
H-SDM	0.244	0.369	0.133
H-SDM (QCM)	0.269	0.393	0.148
H-SDM (Desc)	0.253	0.377	0.138
H-SDM (NX-50, Desc)	0.318	0.440	0.188
H-SDM (everything)	0.322	0.445	0.180

This run is presented by "H-SDM (everything)" in Table 2 and clearly outperforms everything else; excepting the QCM part of it has no significant benefit. In addition to the results reported above, we examined a few issues of preprocessing; we found no effect due to stemming or lemmatization, but found that removing stop-words actually harmed performance.

6 Interactive Methods

In this section we consider the possibility that the person using our algorithm would provide a small amount of information – perhaps indicating which top-level sub-tree is appropriate for the instance being considered, which we consider to be hierarchical relevance feedback, where we consider typical relevance feedback as considering our first 10 results. We expect that while users cannot remember hundreds of nodes in total, a working familiarity with the first level of the hierarchy (See Table 1) will be easier to learn and leverage in an interactive setting.

For each question to be classified, we simulate hierarchy feedback by removing concepts from our ranked list if they are not under the same top-level node in the concept hierarchy as the question. That is, we are simulating the case where a user selects the *correct* top-level category, so any candidates in other sub-trees can be automatically discarded. Table 3 shows that this simple approach ("Hierarchy") provides a substantial gain over no interaction, though it is not as helpful as having the correct question selected from the top 10.

Table 3. Performance with Minimal Feedback.

Feedback	**MRR**	**NDCG**	**P@1**
None	0.318	0.440	0.188
Hierarchy	0.458	0.564	0.287
RF / Success@10	0.598	0.650	0.568
Hierarchy + RF	0.812	0.830	0.781

While the results of this experiment may seem obvious, in lieu of having a user study to determine which of these techniques is easier, quantifying the gains that can be made with this kind of feedback is important. In the case of users familiar with the hierarchy, we expect that we can get a positive gain using both techniques, and for users who are less familiar with the hierarchy (we doubt anyone remembers all 258 leaf nodes), the ranking methods will hopefully provide a much smaller candidate set.

7 Conclusion

In this work we explored the challenge our users face of classifying exam questions into a concept hierarchy, but we explore it from an IR perspective due to the

scarcity of labels available, and our desire to incorporate feedback. This problem was difficult because the exam questions are short and often quite similar and because the concepts in the hierarchy had quite short descriptions. We explored existing approaches, such as document expansion and typical retrieval models, as well as our own methods – especially a hierarchical transform for existing retrieval models that works well, and a model of question structure that provides gains over most baselines.

We hope that our promising results encourage more collaboration between education and information retrieval research, specifically in the identification and exploration of new tasks and datasets that may benefit both fields.

In future work, we hope to explore this problem with other subjects, more exams, and with expert humans in the loop to field-test the feasibility and helpfulness our overall retrieval methods, and our interactive methods.

8 Acknowledgments

The authors thank Prof. Thomas Holme of Iowa State University's Department of Chemistry for making the data used in this study available and Stephen Battisti of UMass' Center for Educational Software Development for help accessing and formatting the data.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant numbers IIS-0910884 and DUE-1323469. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

1. C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.
2. K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Information Processing & Management*, 45(1):1–19, 2009.
3. S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using wikipedia. In *SIGIR '07*, New York, NY, USA, 2007. ACM.
4. R. Bekkerman, H. Raghavan, J. Allan, and K. Eguchi. Interactive clustering of text collections according to a user-specified criterion. In *Proceedings of IJCAI*, pages 684–689, 2007.
5. G. de Melo and G. Weikum. Taxonomic data integration from multilingual wikipedia editions. *Knowledge and information systems*, 39(1):1–39, 2014.
6. S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR '00*, pages 256–263, New York, NY, USA, 2000. ACM.
7. M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short texts through document expansion. In *SIGIR '12*, pages 911–920. ACM, 2012.
8. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.
9. P. Ganesan, H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. *ACM Trans. Inf. Syst.*, 21(1):64–93, Jan. 2003.

10. S. C. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *WWW '06*, pages 633–642. ACM, 2006.
11. T. Holme. Comparing recent organizing templates for test content between ACS exams in general chemistry and AP chemistry. *Journal of Chemical Education*, 91(9):1352–1356, 2014.
12. T. Holme and K. Murphy. The ACS Exams Institute Undergraduate Chemistry Anchoring Concepts Content Map I: General Chemistry. *Journal of Chemical Education*, 89(6):721–723, 2012.
13. K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR 2000*, pages 41–48. ACM, 2000.
14. B. Klimt and Y. Yang. Introducing the enron corpus. In *CEAS*, 2004.
15. J. H. Lee, M. H. Kim, and Y. J. Lee. Information retrieval based on conceptual distance in is-a hierarchies. *Journal of documentation*, 49(2):188–207, 1993.
16. X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR '04*, pages 186–193, New York, NY, USA, 2004. ACM.
17. C. J. Luxford, K. J. Linenberger, J. R. Raker, J. Y. Baluyut, J. J. Reed, C. De Silva, and T. A. Holme. Building a database for the historical analysis of the general chemistry curriculum using ACS general chemistry exams as artifacts. *Journal of Chemical Education*, 2014.
18. D. Metzler and W. Croft. Analysis of statistical question classification for fact-based questions. *Information Retrieval*, 8(3):481–504, 2005.
19. D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR 2005*, pages 472–479. ACM, 2005.
20. N. Omar, S. S. Haris, R. Hassan, H. Arshad, M. Rahmat, N. F. A. Zainal, and R. Zulkifli. Automated analysis of exam questions according to bloom's taxonomy. *Procedia - Social and Behavioral Sciences*, 59(0):297 – 303, 2012.
21. D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. *International Journal on Artificial Intelligence Tools*, 17(01):5–18, 2008.
22. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR 1998*, pages 275–281. ACM, 1998.
23. Z. Ren, M.-H. Peetz, S. Liang, W. van Dolen, and M. de Rijke. Hierarchical multi-label classification of social text streams. In *SIGIR '14*, pages 213–222, New York, NY, USA, 2014. ACM.
24. B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, January 2010. Computer Sciences Technical Report 1648.
25. A. Singhal and F. Pereira. Document expansion for speech retrieval. In *SIGIR '99*, pages 34–41. ACM, 1999.
26. X. Sun, H. Wang, and Y. Yu. Towards effective short text deep classification. In *SIGIR '11*, pages 1143–1144, New York, NY, USA, 2011. ACM.
27. T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *NAACL '06*, pages 407–414. ACL, 2006.
28. P. Wang, J. Hu, H.-J. Zeng, and Z. Chen. Using wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, 19(3):265–281, 2009.
29. G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *SIGIR '08*, pages 619–626, New York, NY, 2008. ACM.
30. D. Zhang and W. S. Lee. Question classification using support vector machines. In *SIGIR '03*, pages 26–32, New York, NY, USA, 2003. ACM.