

Automatic Text Categorization of Mathematical Word Problems

Suleyman Cetintas¹, Luo Si², Yan Ping Xin³, Dake Zhang³, Joo Young Park³

^{1,2}Department of Computer Science, ²Department of Statistics, ³Department of Educational Studies
Purdue University, West Lafayette, IN, 47906, USA

^{1,2}{scetinta, lsi}@cs.purdue.edu, ³{yxin, zhang60, park181}@purdue.edu

Abstract

This paper describes a novel application of *text categorization* for *mathematical word problems*, namely *Multiplicative Compare* and *Equal Group* problems. The empirical results and analysis show that common text processing techniques such as stopword removal and stemming should be selectively used. It is highly beneficial not to remove stopwords and not to do stemming. Part of speech tagging should also be used to distinguish words in discriminative parts of speech from the non-discriminative parts of speech which not only fail to help but even mislead the categorization decision for mathematical word problems. An SVM classifier with these selectively used text processing techniques outperforms an SVM classifier with a default setting of text processing techniques (i.e. stopword removal and stemming). Furthermore, a probabilistic *meta classifier* is proposed to combine the weighted results of two SVM classifiers with different word problem representations generated by different text preprocessing techniques. The empirical results show that the probabilistic meta classifier further improves the categorization accuracy.

Introduction

Math performance on high-stakes tests has become increasingly important in recent years and there has been some improvement in academic achievement following the passage of the No Child Left Behind Act in 2002 (U.S. Dept. of Education 2006). Yet, the overall performance of American students in math has been of particular concern (Gollub, Bertenthal, Labov & Curtis 2002). Increasing trend in computers' utilization for math teaching has led to the development of various intelligent math tutors (Beal 2007; Koedinger et al. 1997; Shiah et al. 1995). As problem solving is the cornerstone of school mathematics (NCTM 2000), there becomes a need for the intelligent tutoring systems to automatically collect mathematical word problems and detect their types before saving them into their problem libraries.

In mathematics education, *word problem* or *story problem* is the term that is often used to refer to any mathematical exercise on which significant background information is presented as text rather than in mathematical

MC	A Formica japonica worker ant weighs 0.004 gram. It can walk while holding in its mouth an object weighing 5 times as much as its own body. How many grams can a worker ant carry?
EG	Darla had 2 singing lessons a month for 2 months. She learned the same number of songs at each lesson. She learned 12 songs in all . How many songs did she learn at each lesson?

Table 1: Multiplicative Compare (MC) and Equal Group (EG) Problem Examples. Note the discriminative stopwords (some of which are also unstemmed) in bold.

notation (Verschaffel 2000). As math word problems are in textual format, detecting their types can be treated as a *text categorization* problem (Sebastiani 2002; Yang 1999).

Automated *text categorization* (TC) is an important technique for organizing text data, which automatically assigns text documents into a fixed number of predefined categories (Sebastiani 2002; Yang 1999). The focus of this paper is a novel application of text categorization for two basic arithmetic word problem types, namely *Multiplicative Compare* (MC) and *Equal Group* (EG) problems (examples given in Table 1) from other types of mathematical word problems. These two problem types are a subset of the most important mathematical word problem types that represent about 78% of the problems in a fourth grade mathematics textbook (Maletsky 2004). This paper studies the classification problem of MC and EG problems from a pool of these important types of mathematical word problems.

Despite a few exceptions (Riloff 1995), prior text categorization research has commonly accepted the fact that text preprocessing methods like stopword removal and stemming often improve the classification accuracy in many applications (Frakes 1992; Porter 1980; Scott 1999; Sebastiani 2002). However for mathematical word problems, this is simply not the case. Some stopwords such as *as*, *than*, *each*, *per* are highly discriminative terms for mathematical word problems since they are strongly associated with different mathematical word problem types. Stemming also hurts the overall performance by transforming terms that are associated with different mathematical word problem types such as *its*, *times* into word stems *it*, *time* which are less helpful for identifying mathematical word problem types. Furthermore, words such as *carry*, *ant* or *learn* in the examples in Table 1 are non-discriminative words and even misleading for

categorization of mathematical word problems: it is helpful to use a part-of-speech (POS) tagger to remove the non-discriminative parts of speech.

In this paper, a Support Vector Machines (SVM) classifier is used for the categorization of mathematical word problems to explore various text preprocessing configurations. Specifically it is shown that an SVM classifier with selectively used text processing techniques (i.e. avoiding stemming, stopword removal and removing non-discriminative parts of speech) outperform an SVM classifier with a default setting of text processing techniques (i.e. stopword removal and stemming).

Furthermore, a probabilistic meta-classifier is built to combine the outputs of SVM classifiers with the two best text preprocessing approaches: in particular (i) the approach that avoids stemming and stopword removal and (ii) the approach that utilizes a part-of-speech (POS) tagger to remove non-discriminative parts of speech while avoiding stemming and stopword removal. The probabilistic meta-classifier consistently improves the categorization accuracy against the individual approaches, demonstrating the power of combining the contributions of the two text-preprocessing configurations.

Related Work

This section briefly surveys common text preprocessing methods and text categorization techniques.

The first well-known text-preprocessing method that is analyzed, stopword removal, suggests that many of the most frequent terms in English such as *like, the, in, each* are not content words because they appear almost in every document, do not carry important information and should be removed (Frakes 1992; Scott 1999; Sebastiani 2002). Stopword lists are commonly used in information retrieval systems (Baeza-Yates 1999).

Another well-known text preprocessing method, stemming or suffix removal, suggests that words that have the same conceptual meaning, such as *computation, computing, compute* should be grouped under the same word stem *comput* (Porter 1980; Scott 1999). Although there are some exceptions (Riloff 1995; Nigam 2000), prior text categorization research has shown that these text processing methods often improve the classification accuracy in many applications (Frakes 1992; Porter 1980; Scott 1999; Sebastiani 2002). However, it will be discussed and analyzed in the later section showing that these commonly-accepted and widely used techniques not only fail to help for the categorization of mathematical word problems but also deteriorate the categorization effectiveness.

To the bests of our knowledge there is no previous work on the categorization of mathematical word problems, so prior work on text categorization is briefed here. Many text categorization methods are proposed in the last two decades (Sebastiani 2002; Yang 1999) such as *k-Nearest Neighbor* approach (kNN) (Dasarathy 1990), the *Naive Bayes* probabilistic classifier (Mitchell 1996), the *Neural*

Network (NNet) method (Dasarathy, 1990) and *Support Vector Machine* (SVM) classifier. Space limitations preclude discussing them all, so only the SVM classifier that is used in this work is explained.

Support Vector Machine (SVM) is a learning approach that is based on structural risk minimization principle. The problem SVM deals with is to find a decision surface that best separates the data points in two classes over a vector space. SVM has been shown to achieve state-of-the-art generalization performance over a wide variety of application domains of which space limitations preclude mentioning and is one of the most accurate and widely used text categorization techniques (Joachims 1998; Yang 1999). Therefore SVM is used as the TC method in this work and particularly the simplest linear version of SVM is chosen since it is fast to learn and fast to classify new instances (Joachims 1998; Yang 1999).

Methods

This section first presents various configurations of text preprocessing techniques that are used with SVM classifiers for the categorization task and then proposes a probabilistic meta-classifier to combine the weighted results from two best configurations.

Text Preprocessing Approaches

Avoiding Stopword Removal and Stemming. Many stopwords are strongly associated with different mathematical word problem types. For instance, *as & times* in expressions like *five times as much as* are good indicators of an MC problem, likewise *each* in expressions like *each lesson* and *in all* in expressions like *12 songs in all* are pointing out an EG problem (see Table 1). Therefore the commonly accepted text preprocessing method of removing stopwords (Frakes 1992; Scott 1999; Sebastiani 2002) should be avoided for classification of mathematical word problems. As the first preprocessing method in this paper, the effect of avoiding stopword removal with an SVM classifier that uses a linear kernel is investigated and referred as SVM-NSTOP.

Another widely used text preprocessing method, stemming or suffix removal, suggests that word suffixes are often important for text categorization and that words with the same conceptual meaning, such as *computation, computing, compute* should be grouped under the same word stem *comput* (Porter 1980; Scott 1999). However for the categorization of mathematical word problems, stemming hurts the categorization performance as it transforms some of the words that are associated with different mathematical word problems into word stems that can exist in any mathematical word problem type. For instance, *its, times* that are highly associated with MC problems are transformed into word stems *it, time* that no longer indicate MC problems (see Table 1). The natural conclusion from this observation is to avoid stemming and use the original form of the words (i.e. without suffixes

removed). Therefore the effect of avoiding stemming over categorization accuracy is investigated next and referred as SVM-NSTEM.

To more thoroughly study the effect of stemming over the categorization effectiveness while stopword removal is avoided, a corresponding configuration is utilized as SVM-NSTOP-NSTEM.

The Lemur information retrieval toolkit¹ was used for stemming and stopword removal. The INQUERY stopwords list (Callan 1995) and Porter stemmer (Porter 1980) were utilized to do stopword removal and stemming respectively.

Part of Speech Tagging (POS) Utilization. For categorization of mathematical word problems, not all parts of speech are discriminative. In particular some words such as *worker, ant, singing, lesson, math, homework, highway, speed* etc. in the example problems given in Table 1 are not associated with any mathematical word problem type, i.e. they can exist in any mathematical word problem type, so they are non-discriminative. These kinds of non-discriminative words not only fail to discriminate the mathematical word problems but also make the classification task more complicated by misleading the classifiers. In particular, when non-discriminative words are distributed nonuniformly across problem types in training and test problems, these words make the classifier biased towards one or more types and result in misleading classification decisions. Therefore it is important to distinguish the discriminative parts of speech from the non-discriminative ones and eliminate the non-discriminative parts of speech before a classification algorithm is applied on the mathematical word problems.

Another issue is that although some stopwords are extremely discriminative for mathematical word problems as discussed before, some other stopwords are misleading such as *he, she, it, is, are, how, many, there* etc. Distinguishing indicative stopwords from the non-indicative ones rather than blindly letting every stop-word reside in the mathematical word problems is of particular interest. Selecting the discriminative parts of speech from the non-discriminative ones and eliminating the non-discriminative parts of speech will also be important to differentiate the indicative stopwords from the non-indicative ones before a classification algorithm is applied on the mathematical word problems. Therefore a set of parts of speech that are thought to cover most –if not all– non-indicative parts of speech and stopwords that give almost no information about the type of mathematical word problems are selected and removed from math word problems and then the classification algorithms are run over those carefully preprocessed problems. Particularly a part-of-speech (POS) tagger² is used to eliminate nouns (tagged as NN, NNP, NNPS, NNS), verbs (VB, VBD, VBG, VBN, VBP, VBZ), modal auxiliaries (MD), Wh-

Table 2: The number of average terms for each problem type after stemming and stopword removal under the *default* column; after avoiding stopword removal under the *with stopwords* column and after non-discriminative word types are eliminated with POS tagger under the *after POS tg.* column.

Problem Types	# of Problems	Average Length (words)		
		default	with stopwords	after POS tg.
MC	60	18	31	10
EG	60	12	23	7
Other	112	15	28	10

adverbs (WRB), adjective or numeral, ordinals (JJ), numeral, cardinals (CD), existential there (EX) and interjections (UH) that are all non-discriminative.

The non-discriminative parts of speech are eliminated while stemming and stopword removal are avoided and this approach is referred as SVM-NSTOP-NSTEM-POS.

Statistical Learning Methods

A Support Vector Machines (SVM) classifier is utilized to explore different configurations of text-preprocessing techniques. Furthermore, a probabilistic meta-classifier is proposed to further improve the accuracy of the mathematical word problem classification task by combining the contributions of two best text preprocessing methods.

Support Vector Machine. Support Vector Machines (SVM) have been shown to provide state-of-the-art generalization performance over a wide variety of application domains and is also one of the most accurate as well as widely used text categorization techniques (Joachims 1998; Yang 1999). In this work the simplest linear version of SVM (i.e. with a linear kernel) was used as the TC classifier³ that can be formulated as a solution to an optimization problem as follows:

$$\begin{aligned} \{\vec{w}, b\} = \arg \max_w \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to: } y_i (\vec{w} \bullet \vec{d}_i + b) - 1 + \xi_i \geq 0 \quad \forall \xi_i \geq 0 \end{aligned} \quad (1)$$

where \vec{d}_i is the i document represented as a bag of words vector in the TC task; $y_i \in \{\pm 1\}$ is the binary classification of \vec{d}_i ; \vec{w} , b are the parameters of the SVM model. C has the control over the tradeoff between classification accuracy and margin, which is tuned empirically. The categorization threshold of each SVM classifier is learned by 5-fold cross validation in the training phase.

Probabilistic Meta Classifier. Meta-classifiers endeavor to improve the accuracy of a classification task by combining the results of a set of individual classifiers with appropriate weights and making a final decision out of individual classifiers' results in a weighted way. So the

¹ <http://www.lemurproject.org/>

² <http://nlp.stanford.edu/software/tagger.shtml>

³ the SVM^{light} toolkit (Joachims 1999) was utilized.

judgments/scores from each classifier for each class are treated as a feature and the meta classifier is built to make the final decision (Lin 2002). In this work, a probabilistic meta classifier is designed to combine the decisions of SVM-NSTOP-NSTEM and SVM-NSTOP-NSTEM-POS classifiers by learning their weights through 5-folds cross validation to estimate whether a particular problem is of a particular type as follows:

$$P(y=1 | \beta_0, \beta_1, \beta_2, S_{npos}, S_{pos}) = \frac{\exp(\beta_0 + \beta_1 S_{npos} + \beta_2 S_{pos})}{1 + \exp(\beta_0 + \beta_1 S_{npos} + \beta_2 S_{pos})} \quad (2)$$

where $y=1$ indicates that the problem belongs to the particular type, S_{npos} is the score from SVM-NSTOP-NSTEM classifier for the problem and β_1 is the weight of SVM-NSTOP-NSTEM classifier, S_{pos} is the score from SVM-NSTOP-NSTEM-POS classifier for the problem and β_2 is the weight of SVM-NSTOP-NSTEM-POS classifier.

The main motivation to build the meta-classifier is to balance the contributions of the two best approaches, i.e. the approach eliminating the non-discriminative parts of speech while avoiding stemming and stopword removal (i.e. SVM-NSTOP-NSTEM-POS) and the approach that only avoids stemming and stopword removal (i.e. SVM-NSTOP-NSTEM) and successfully combine their advantageous sides.

The proposed meta classifier, which is referred as PR-COMB-SVMs, achieves better or at least as good as the performance of the individual approaches by combining the contribution of each method as shown in Experiment Results section.

Experimental Methodology

Dataset

To the best of our knowledge, there is no previous work using math word problem corpora so far, therefore we built our own corpora for our experiments. Hence 232 mathematical word problems are manually (i.e. by hand) collected from Grade 3-5 mathematics textbooks (Maletsky 2004) under the guidance and with the help of our collaborators who are experts in educational studies. Along with MC and EG problems, some other types of mathematical word problems are also used so that the classifiers can not only differentiate MC and EG problems from each other, but also from other important mathematical problem types. As shown in Table 2, a total of 60 MC, 60 EG, 112 other types of mathematical word problems are used all together. The details about the average length of each problem type (i) after stopwords' elimination, (ii) after avoiding stopwords' elimination and (iii) after eliminating the non-discriminative parts of speech (while avoiding stopword removal) are given in Table 2. Note that stemming doesn't change the length of problems.

Baseline SVM Classifier

In the experiments, the methods that are proposed in Methods section are compared to a standard SVM classifier as a text categorization method (with stemming and stopword removal applied on the data as they are almost always used for the applications of SVMs to text categorization). SVM classifier serves as a good baseline since it has been shown to provide state-of-the-art generalization performance over a wide variety of application domains and is one of the most accurate and widely used text categorization techniques (Joachims 1998; Yang 1999) as mentioned in Support Vector Machine section. This baseline SVM classifier will be referred as SVM-BASELINE.

Evaluation Metric

To evaluate the effectiveness of the categorization of the word problems, the common F_1 measure that is the harmonic mean of precision and recall is used (Rijsbergen 1979). Precision (p) is the ratio of the correct categorizations by a classifier divided by all the categorizations of that classifier. Recall (r) is the ratio of correct categorizations by a classifier divided by the total number of correct categorizations.

Experiment Results

This section presents the experimental results of the methods that are proposed in Methods section. All the methods were evaluated on the dataset as described in Dataset section.

An extensive set of experiments are conducted to address the following questions:

- How effective are the TC methods that avoid stopword removal (i.e. SVM-NSTOP), stemming (SVM-NSTEM) and both of stemming and stopwords' elimination (SVM-NSTOP-NSTEM)?
- How effective is the TC method that avoids stemming, stopwords' elimination and utilizes a part of speech tagger (SVM-NSTOP-NSTEM-POS)?
- How effective is the combined TC method (PR-COMB-SVMs)?

Effectiveness of Avoiding Stopword Removal and Stemming

The first set of experiments was conducted to evaluate the effectiveness of the text preprocessing methods of avoiding stemming and stopword removal. The details about these approaches are given in detail in Avoiding Stopword Removal and Stemming section.

More specifically, SVM-NSTOP, SVM-NSTEM and SVM-NSTOP-NSTEM classifiers are compared with SVM-BASELINE classifier on both MC and EG categorization tasks. The performance of SVM-NSTOP, SVM-NSTEM and SVM-NSTOP-NSTEM classifiers are

Table 3: The results of SVM-NSTOP, SVM-NSTEM, SVM-NSTOP-NSTEM, SVM-NSTOP-NSTEM-POS and PR-COMB-SVMs classifiers are shown in comparison to SVM-BASELINE classifier for the categorization of MC and EG problems. The percentages in the parenthesis show the relative improvements of each method with respect to the SVM-BASELINE classifier. The performance is evaluated by the F_1 measure.

Methods	MC Problems	EG Problems	Average
SVM-BASELINE	0.696	0.565	0.630
SVM-NSTEM	0.630 (-10.4%)	0.528 (-07.1%)	0.579 (-08.9%)
SVM-NSTOP	0.712 (+02.2%)	0.781 (+27.6%)	0.746 (+15.5%)
SVM-NSTOP-NSTEM	0.756 (+07.9%)	0.785 (+28.0%)	0.775 (+18.7%)
SVM-NSTOP-NSTEM-POS	0.759 (+08.3%)	0.811 (+30.3%)	0.785 (+19.6%)
PR-COMB-SVMs	0.780 (+10.7%)	0.811 (+30.3%)	0.795 (+20.7%)

shown in comparison to SVM-BASELINE classifier in Table 3. It can be seen that the SVM-NSTOP classifier substantially outperforms SVM-BASELINE classifier. This explicitly demonstrates the discriminative power of some stopwords for the categorization of mathematical problems (i.e. MC and EG problems). SVM-NSTEM classifier performs much worse than SVM-BASELINE classifier. This is due to the fact that most words that favor the technique of avoiding stemming are stopwords and avoiding stemming only shows its power when stopwords are not removed. The next row in Table 3 shows the performance of SVM-NSTOP-NSTEM classifier which outperforms SVM-BASELINE, SVM-NSTEM and SVM-NSTOP classifiers. This is due to the following facts: (i) stopwords are strongly associated with different mathematical word problem types and should not be removed; (ii) most of the words that are favored by avoiding stemming are stopwords and without stopwords, avoiding stemming doesn't help to improve the categorization effectiveness; (iii) avoiding stemming successfully improves the categorization accuracy when it is used with avoiding stopword removal. It can also be observed from Table 3 that SVM-NSTOP and SVM-NSTOP-NSTEM classifiers have bigger effectiveness improvements over the SVM-BASELINE classifier for EG problems than they have for MC problems. This is due to the fact that stopwords associated with EG problems (e.g. “each”) are more discriminative than their counterparts for MC problems. This fact also leads to the different trends in performances of the SVM-NSTOP-NSTEM-POS and PR-COMB-SVMs classifiers for EG and MC. One key lesson learned from the set of experiments in this section is that text preprocessing methods should be used with special care rather than blindly using or avoiding commonly accepted techniques since utilization of one technique (e.g. avoiding stemming depends on avoiding stopword removal in this piece of work) can also depend on the utilization of another technique as explained here.

Effectiveness of Removing Nondiscriminative Parts of Speech

The second set of experiments was conducted to evaluate the effectiveness of the text preprocessing methods of eliminating the non-discriminative parts of speech while avoiding stemming and stopword removal. The details

about this approach are given in detail in Part of Speech Tagging (POS) Utilization section.

Particularly SVM-NSTOP-NSTEM-POS classifier is compared with SVM-BASELINE and SVM-NSTOP-NSTEM classifiers. The performance of SVM-NSTOP-NSTEM-POS classifier is shown in Table 3. It can be seen that the SVM-NSTOP-NSTEM-POS classifier substantially outperforms SVM-BASELINE classifier and also improves the categorization effectiveness over SVM-NSTOP-NSTEM classifier on both MC and EG categorization tasks. This result suggests that differentiating non-discriminative parts of speech from the discriminative ones as well as differentiating non-discriminative stopwords from the discriminative ones and removing all the non-discriminative parts of speech (and also non-discriminative stopwords) is highly important for the categorization effectiveness of mathematical word problems.

Effectiveness of Combining Classifiers

The third set of experiments was conducted to evaluate the effectiveness of the probabilistic meta classifier that combines the outputs of two best text preprocessing approaches: in particular the approach that avoids stemming and stopwords removal (i.e. SVM-NSTOP-NSTEM) and the approach that avoids stemming, stopwords removal while eliminating the non-discriminative parts of speech (i.e. SVM-NSTOP-NSTEM-POS). The details about this approach are given in detail in Probabilistic Meta Classifier section.

This section specifically compares PR-COMB-SVMs classifier with SVM-NSTOP-NSTEM and SVM-NSTOP-NSTEM-POS classifiers. The performance of PR-COMB-SVMs classifier in comparison to SVM-NSTOP-NSTEM, SVM-NSTOP-NSTEM-POS and SVM-BASELINE classifiers are shown in Table 3. It can be seen that the PR-COMB-SVMs classifier substantially outperforms SVM-BASELINE classifier on both MC and EG categorization tasks and performs remarkably better than (or at least as much as) SVM-NSTOP-NSTEM or SVM-NSTOP-NSTEM-POS classifiers. This explicitly shows the power of combining the contributions of the two text preprocessing configurations. The lesson to learn from this set of experiments is that the approach of eliminating the non-discriminative parts of speech (i.e. SVM-NSTOP-

NSTEM-POS classifier) sometimes eliminates too much that almost nothing is left from a mathematical word problem for the classifier to make its decision. Hence it becomes beneficial to combine the SVM-NSTOP-NSTEM-POS approach along with an alternative approach for compensation and SVM-NSTOP-NSTEM, as the best performing alternative, makes PR-COMB-SVMs the best performer among all classifiers that have been explored.

Conclusions and Future Work

This paper describes a novel application of text categorization for mathematical word problems. The effects of several text preprocessing approaches are studied for this application. Several significant conclusions are observed from the empirical results. In short, the results show that by avoiding commonly used text preprocessing techniques such as stemming and stopword removal and using a POS tagger to eliminate non-discriminative words, an SVM classifier can outperform another SVM classifier with a default setting of text processing techniques (i.e. with stopword removal and stemming). Furthermore, a probabilistic meta-classifier is proposed to combine the outputs of two SVM classifiers with different word problem representations for eliminating the downsides of individual classifiers and successfully combining their advantages. It is shown that the meta classifier successfully improves the classification accuracy of mathematical word problems.

There are several possibilities to extend the research. The SVM classifiers in this work are using single words as their features (i.e. unigram model): it may be helpful to explore n-gram models or a linear interpolation of unigram and n-gram models for smoothing. It is also an interesting direction to feed the SVM classifiers with additional features such as phrasal or pseudo regular expressions. It's possible to design a more sophisticated probabilistic meta classifier that models the combination of these new directions with the approaches this paper proposes to further improve the classification accuracy.

Acknowledgements

This research was partially supported by the NSF grants IIS-0749462, IIS-0746830 and DRL-0822296. Any opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsor.

References

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. *ACM Press Series/Addison Wesley*.
Beal, C.R. (2007). On-line Tutoring for Math Achievement Testing: A Controlled Evaluation. *Journal of Interactive Online Learning*, 6(1):43-55.

Callan, J., Croft W. B. & Broglio, J. (1995). TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31(3).
Dasarathy, B. V. (1990). Nearest neighbor (NN) norms: NN pattern classification techniques. In *Los Alamitos: IEEE Computer Society Press, 1990*.
Frakes, W., & Baeza-Yates, R. (1992). Information Retrieval: Data Structures and Algorithms. *Prentice Hall, Englewood Cliffs, NJ*.
Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*
Joachims, T. (1999). Making large-scale support vector machine learning practical. *MIT Press*.
Koedinger, K. R., Anderson, J.R., Hadley, W.H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
Lin, W. & Hauptmann, A. (2002). News video classification using svm-based multimodal classifiers and combination strategies. In *ACM Multimedia-2002, 10th ACM International Conference on Multimedia*
Maletsky, E. M. et al. (2004). Harcourt Math (Indiana Edition). *Chicago: Harcourt*.
Mitchell, T. (1996). Machine Learning. *McGraw Hill*.
Nigam, K., Mcallum, A. K., Thrun, S. and Mitchell T. M. (2000). Text classification from labeled and unlabeled documents using EM. In *Mach. Learn.* 39, 2/3, 103–134.
National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: National Council of Teachers of Mathematics.
Porter, M. F. (1980). An algorithm for suffix stripping. In *Program: Electronic Library and Information Systems*, 14(3):130–137.
Rijsbergen, C. J. v. (1979). Information Retrieval, 2nd edition. *University of Glasgow*.
Riloff, E. (1995). Little words can make a big difference for text classification. In *Proceedings of ACM SIGIR-95, 18th ACM SIGIR Conference on Research and Development in Information Retrieval*, 130–136.
Scott, S. & Matwin, S. (1999). Feature engineering for text classification. In *Proceedings of ICML-99, 16th International Conference on Machine Learning*.
Shiah, R., Mastropieri, M. A., Scruggs, T.E., & Fulk, B.M. (1995). The effects of computer-assisted instruction on the mathematical problem solving of students with learning disabilities. *Exceptionality*, 5, 131-161.
U.S. Dept. of Education. (2006). No Child Left Behind is working. Retrieved November, 2008, from <http://www.ed.gov/nclb/overview/importance/nclbworking.html>
Verschaffel, L., Greer, B., Corte, D. E. (2000). Making Sense of Word Problems. *Taylor & Francis*
Yang, Y. & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, 42–49.