Proceedings of the Twenty-Eighth AAAI Conference on Innovative Applications (IAAI-16)

MetaSeer.STEM: Towards Automating Meta-Analyses

Kishore Neppalli,[1] **Cornelia Caragea,**[1] **Robin Mayes,**[2] **Kim Nimon,**[3] **Fred Oswald**[4]
[1]Computer Science, University of North Texas, Denton
[2]Learning Technologies, University of North Texas, Denton
[3]Learning Technologies, University of Texas, Tyler
[4]Department of Psychology, Rice University

Abstract

Meta-analysis is a principled statistical approach for summarizing quantitative information reported across studies within a research domain of interest. Although the results of meta-analyses can be highly informative, the process of collecting and coding the data for a meta-analysis is often a labor-intensive effort fraught with the potential for human error and idiosyncrasy. This is due to the fact that researchers typically spend weeks poring over published journal articles, technical reports, book chapters and other materials in order to retrieve key data elements that are then manually coded for subsequent analyses (e.g., descriptive statistics, effect sizes, reliability estimates, demographics, and study conditions). In this paper, we propose a machine learning based system developed to support automated extraction of data pertinent to STEM education meta-analyses, including educational and human resource initiatives aimed at improving achievement, literacy and interest in the fields of science, technology, engineering, and mathematics.

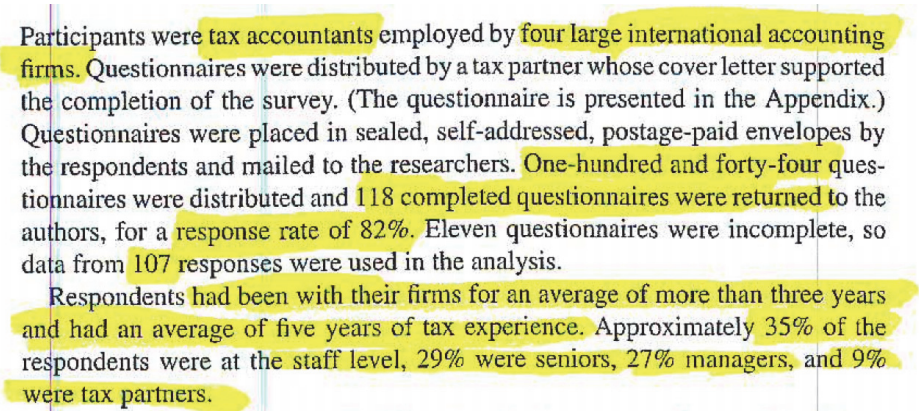
Introduction

As science advances, scientists around the world continue to produce large numbers of research articles. Meta-analysis is a statistical process for summarizing quantitative information reported across such articles within a research domain of interest (Schmidt 2008; Schmidt and Hunter 1977; Vacha-Haase 1988). In the last three decades, meta-analysis has been used widely across scientific disciplines to answer both theoretical and practical questions. In STEM education, meta-analysis, for example, has summarized the body of literature examining gender differences in mathematics performance (Lindberg, Hyde, and Petersen 2010; Hyde et al. 2008; Lindberg, Hyde, and Hirsch 2008) and the use of technology in postsecondary education (Schmid et al. 2014). After examining about 4,000 studies published between 1990 and 2007 and finding 242 studies with sufficient statistical information to calculate effect sizes, Lindberg et al. (2010) found virtually no gender differences in mathematics performance, confirming an earlier meta-analysis conducted by Hyde, Fennema, and Lamon (1990). After examining 1,105 full-text documents, Schmid et al. (2014) found 870 usable effect sizes derived from 674 studies that when analyzed indicated that the use of technology in post-secondary education classrooms yielded positive effects in terms of scholastic achievement and attitude outcomes.

As illustrated by these examples, meta-analyses typically require sifting through large document sets, e.g., about 4,000 documents in Lindberg et al. (2010), and about 1,100 documents in Schmid et al. (2014), to arrive at reliable conclusions. Figure 1 shows an example of a small amount of text manually coded for a meta-analysis. The fragment of text is taken from an article by Luttman, Mittermaier, and Rebele (2003). Although a team of researchers is helpful to code the documents, it presents critical issues regarding the nature and reliability of the coding process. Although psychometric checks for inter-coder agreement are critical, imagine the very real problem of critical pieces of information that were not gathered or not gathered consistently across research members. Furthermore, coders can share the same blind spots or opinions of the data, where they may mistakenly agree on the wrong pieces of information that are present or absent. In other words, statistical indices of agreement will not capture all the inconsistencies or mistaken consistencies in comparing codes. The process of manually coding data elements from a corpus of articles into a spreadsheet provides no support for: (a) resolving conflicts that arise when assessing inter-rater reliability or (b) answering questions that arise when cleaning the data, other than going back to the original article.

Hence, due to the high cost and effort involved in coding the data for a meta-analysis fraught with the potential for human error, there is a growing interest in automated methods that can *efficiently* and *effectively* retrieve key data elements that are used for subsequent analyses (e.g., descriptive statistics, effect sizes, reliability estimates, demographics, and study conditions).

Machine learning offers a promising approach to the design of algorithms for training computer programs to accurately extract relevant data from research articles. One of the main challenges in learning from such data is to filter from the huge amount of information available in these documents the small fraction of key data elements required for meta-analyses. Thus, one question that can be raised is the following: *Can we design robust automated approaches for the extraction of relevant data for meta-analyses, in order to minimize the human effort involved in coding large*



Participants were tax accountants employed by four large international accounting firms. Questionnaires were distributed by a tax partner whose cover letter supported the completion of the survey. (The questionnaire is presented in the Appendix.) Questionnaires were placed in sealed, self-addressed, postage-paid envelopes by the respondents and mailed to the researchers. One-hundred and forty-four questionnaires were distributed and 118 completed questionnaires were returned to the authors, for a response rate of 82%. Eleven questionnaires were incomplete, so data from 107 responses were used in the analysis.

Respondents had been with their firms for an average of more than three years and had an average of five years of tax experience. Approximately 35% of the respondents were at the staff level, 29% were seniors, 27% managers, and 9% were tax partners.

Figure 1: Example excerpt from an article with highlighted key data elements useful for Meta-Analysis.

amounts of data for successful meta-analyses? In this paper, we specifically address this question with our research agenda.

Contributions. We propose and develop a reusable information technology system called MetaSeer.STEM that allows STEM education researchers to automatically extract key data elements from a corpus of articles, which can subsequently be used in a meta-analysis task. More precisely, MetaSeer.STEM allows the automatic extraction and classification of variables of interest and their values from research articles so that they can be easily accessed and used by researchers for various meta-analysis tasks. MetaSeer.STEM has the potential to make meta-analysis available to a wide range of applied STEM education researchers, much like tools such as SPSS[1] and SAS[2] have made statistical analyses available to a broad range of researchers. MetaSeer.STEM is designed so that it is applicable to other domains of research including management, psychology, sociology, political science, statistics, and health informatics. Although in a proof-of-concept stage, the results of our experiments show that MetaSeer.STEM shows great potential as a tool for meta-analysis tasks.

The rest of the paper is organized as follows. In the next section, we review some of the related work. We present our system and its main components in the "Information Extraction for Meta-Analyses" section. We then describe the system evaluation and discussion of results before concluding the paper.

Related Work

In a review of literature related to information extraction of meta-data, we found data-mining tools that were specifically designed for the medical science researchers, including clinical trial researchers and neuroscientists. Restificar and Ananiadou (2012) designed a method for discovering and inferring appropriate eligibility criteria in clinical trial applications without labeled data. Kakrontzelos et al. (2011) introduced ASCOT that uses text mining and data mining methods to extract a pre-defined set of clinical trial parameters (e.g., study type, study design, gender) from large clinical trial collections. Similarly, ExACT (Kiritchenko et al. 2010) automatically extracts 21 pre-defined clinical trial characteristics from journal publications. Still other tools have been developed to support data mining for neuroscientists. Nielsen, Balslev, and Hansen (2005) designed a method to mine neuroimage database to extract the main functional modules within a brain image. Yarkoni et al. (2011) designed software that enables neuroscientists to search online articles for various keywords (e.g., pain) and automatically extract brain coordinates from the articles' tables. Despite these advances in the medical research field, no apparent advances have been provided to support the meta-analysis of STEM education research or other social science research. Wu et al. (2014) presented an overview of AI technologies used in the CiteSeerX digital library for several tasks such as document classification, de-duplication, meta-data extraction, citation extraction, table/figure search and author disambiguation. Our work is in line with the meta-data extraction task. We particularly focus on the extraction of relevant numbers from the whole text of a document for meta-analyses.

Information Extraction for Meta-Analyses

We propose MetaSeer, a system that allows automated data extraction from a corpus of articles needed to conduct a meta-analysis. We first describe the system architecture and then present the details of the main components of MetaSeer.

System Architecture

The MetaSeer system architecture is shown in Figure 2 and includes three main components: (1) MetaSeer Data Extractor, (2) MetaSeer Trainer, and (3) MetaSeer Classification. The MetaSeer Data Extractor extracts the text from the pdf documents, parses the text and returns all numbers. The MetaSeer Trainer trains a classifier from a labeled set of examples. The trained classifier is further used by the MetaSeer Classification component to identify numbers of interested from all numbers extracted from a document.

The typical use case for the MetaSeer system follows:

1. A meta-analysis document corpus is presented to the system by a user (or a team of users) interested in a particular

[1] http://www-01.ibm.com/software/analytics/spss/

[2] http://www.sas.com/en_us/home.html

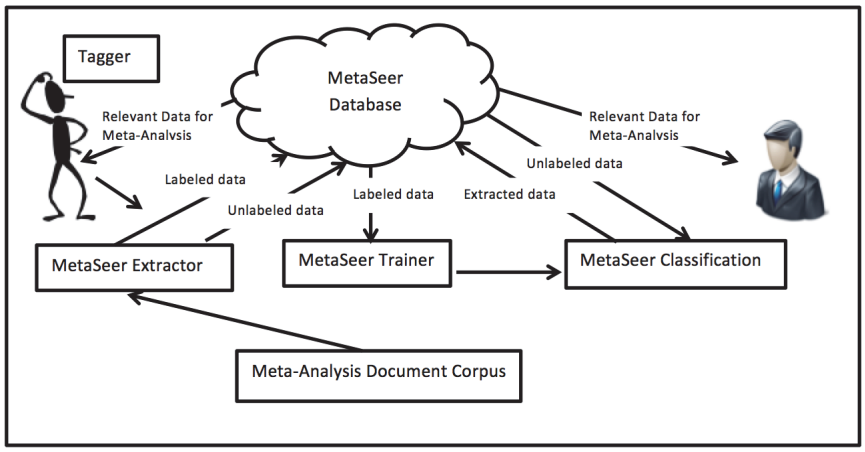


Figure 2: MetaSeer.STEM System Architecture.

meta-analysis task. The documents in the corpus can be either pdf or text documents. For pdf documents, a supported pdf to text conversion tool (e.g., PDFBox[3]) is first used to extract the text from the pdf.

2. The system extracts all numbers from the text of the documents in the corpus and presents them to the user, who manually annotates these numbers (or a fraction of them) as numbers of interest vs. numbers of non-interest for the task. For example, a user annotates numbers of interest (i.e., the positive examples) by a variable name (e.g., sample size, response rate) and a variable type (e.g., mean, standard deviation, count) and numbers of non-interest (i.e., the negative examples) as everything else. Note that the MetaSeer Data Extractor automatically archives the data that is not needed for the meta-analysis without action from the user (e.g., years that occur in the text of the documents or page numbers). The result of this step is a labeled dataset of examples, which are stored in the MetaSeer database. Users can directly use the coded numbers for meta-analysis, or automated data extraction methods can be further developed.
3. The labeled dataset is pushed to the MetaSeer Trainer, where a classifier is trained to discriminate between numbers of interest vs. numbers of non-interest. A subset of the labeled dataset is used for training and an independent subset is used for testing to evaluate the classifier.
4. If the user is satisfied with the performance of the trained classifier, then the classifier can be used by the MetaSeer Classification component to automatically annotate more unlabeled data for better meta-analysis in order to minimize human effort. The relevant key data elements are stored again in the MetaSeer database. If the classifier's performance is unacceptable, the user is asked to label more data, and Step 3 above is repeated.
5. Finally, the relevant data are retrieved from the database and presented to the user to perform a particular meta-analysis task. Note that the database can store data from various domains, e.g., mathematics or music.

[3]http://pdfbox.apache.org/

Next, we present details of the main MetaSeer components.

MetaSeer Data Extractor

This component takes as input a document corpus and extracts numbers represented as numerals as well as numbers that are expressed as words from the text of the documents in this corpus input by a user (i.e., a meta-analyst). The extracted numbers are presented to the user through a user interface. A screenshot of this interface is shown in Figure 3. The left panel contains the list of documents to be analyzed, whereas the right panel shows the current number that is extracted along with its *context*, i.e., a window of n characters on each side of the number as they appear in the document's text. The value of n is a user input parameter, set to 250 (see Figure 3). The user manually annotates the current number as relevant with the variable name and variable type or as non-relevant and can choose to move to the next extracted number or to stop. To minimize the quantity of numbers that a user has to respond to, the application automatically skips over: (a) numbers that appear to be years in citations, e.g., 2004 in (Schmidt & Hunter, 2004); (b) numbers that appear to be page numbers in citations, e.g., 530 in (Schmidt & Hunter, 2004, p. 530); (c) numbers that occur before the Method section (when indicated by the user); and (d) numbers that occur after the References section. The component also allows the user to complete the annotation process for a given document by selecting the "Cancel" button. Any numbers that are not presented to the user are automatically saved and annotated in the labeled dataset as not having been selected by the user due to the particular reason (e.g., year, page number, before Method, in References).

The output of this component is a dataset of examples, i.e., numbers along with their contexts. The examples can be manually labeled by a user as relevant or non-relevant, or can be unlabeled examples (coming from documents that are not manually labeled by the user). Each example labeled as relevant will reflect one of the variables to be coded in the meta-analysis, such as response rate or percentage of males

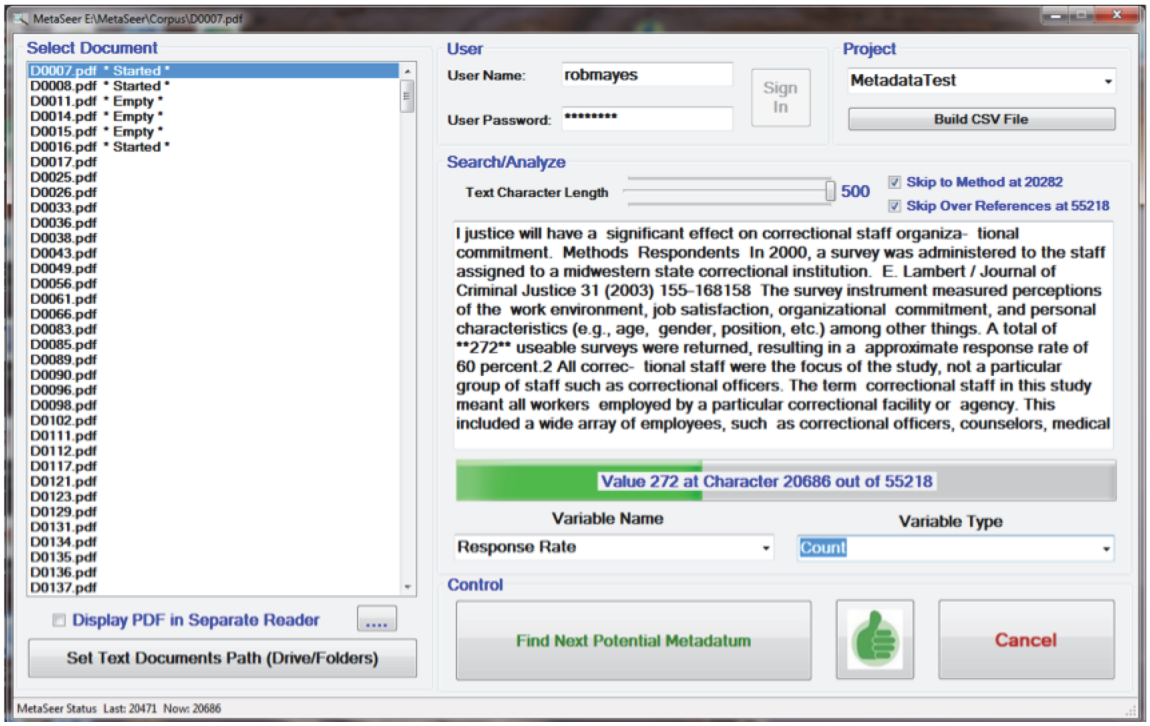


Figure 3: MetaSeer Data Extractor screenshot.

in the sample. Each example labeled as non-relevant represents a number that is irrelevant for meta-analysis.

MetaSeer Trainer and Classification

The Trainer component takes as input the labeled dataset returned by the Extractor. This dataset is used to train classifiers in a supervised learning fashion in order to automate the process of coding for meta-analysis.

The supervised learning problem (Bishop 2006; Mitchell 1997) can be formally defined as follows: Given an *independent and identically distributed* (*iid*) data set $\mathcal{D}$ of labeled examples $(\mathbf{x}_i, y_i)_{i=1,\cdots,n}$, $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the set of all possible class labels; a hypothesis class $\mathcal{H}$ representing the set of all possible hypotheses that can be learned; and a performance criterion P (e.g., accuracy), a learning algorithm L outputs a hypothesis $h \in \mathcal{H}$ (i.e., a classifier) that optimizes P. The input $\mathbf{x}_i$ can represent natural text over a finite vocabulary of words $\mathcal{X}$, $\mathbf{x}_i \in \mathcal{X}^*$. During classification, the task of the classifier h is to accurately assign a new example $\mathbf{x}_{test}$ to a class label $y \in \mathcal{Y}$ (see Figure 4). In our case, examples are numbers and their contexts and the set of class labels is $\mathcal{Y} = \{+1, -1\}$, corresponding to relevant and irrelevant numbers, respectively.

The feature representation used to encode our data is the "bag of words" representation, which is widely used by the machine learning community for many classification tasks (see, for instance, (McCallum and Nigam 1998)). Each example is drawn from a multinomial distribution of words from a vocabulary, and the number of independent trials is equal to the length of the example.The "bag of words" approach first constructs a vocabulary of size d, which contains all words in each context in the corpus of documents. A context is represented as a vector $\mathbf{x}$ with as many entries as the words in the vocabulary, where an entry k in $\mathbf{x}$ can record the frequency (in the context) of the k^{th} word in the vocabulary, denoted by x_k. Because only a small number of words (compared to the vocabulary size) occurs in a context, the representation of $\mathbf{x}$ is very sparse, i.e., only a small number of entries of $\mathbf{x}$ is non-zero. In our implementation, we used sparse representations of contexts. In experiments, we trained Naïve Bayes classifiers using the "bag of words" representation.

The Naïve Bayes classifier (Mitchell 1997) is one of the simplest probabilistic approaches. It belongs to the class of generative models, in which the probabilities $p(\mathbf{x}|y)$ and $p(y)$ of the input $\mathbf{x}$ and the class label y are estimated from the data. In general, the input $\mathbf{x}$ is high-dimensional, represented as a tuple of attribute values, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$, making it impossible to estimate $p(\mathbf{x}|y)$ for large values of n. However, Naïve Bayes classifier makes the strong independence assumption that the attribute values are conditionally independent given the class. Therefore, training a Naïve Bayes classifier reduces to estimating probabilities $p(\mathbf{x}_i|y)$, $i = 1, \cdots, n$, and $p(y)$. During classification, Bayes Rule is applied to compute $p(y|\mathbf{x}_{test})$. The class label with the highest posterior probability is assigned to the new input $\mathbf{x}_{test}$.

In the MetaSeer Trainer component, the trained classifiers are evaluated using k-fold cross-validation. K-fold cross-validation is an evaluation scheme considered by many authors to be a good method of estimating the *generalization*

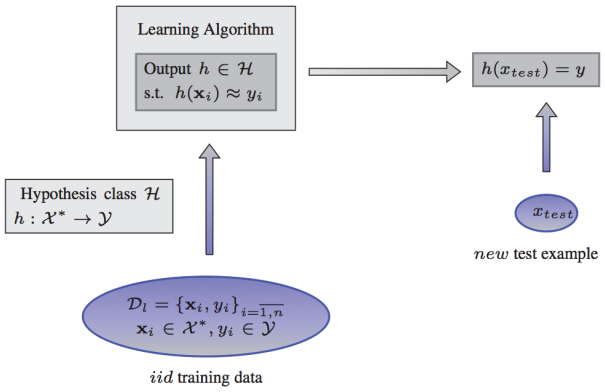


Figure 4: Supervised Learning and Classification.

accuracy of a predictive algorithm (i.e., the accuracy with which the predictive algorithm fits examples in the test set). This evaluation scheme can be described as follows: the original dataset $\mathcal{D}$ containing n instances is randomly partitioned into k disjoint subsets of approximately equal size, $\approx n/k$. The cross-validation procedure is then performed k different times. During the i^{th} run, $i = 1, \cdots, k$, the i^{th} subset is used for testing and the remaining $k-1$ subsets are used for training. Therefore, each instance in the dataset is used exactly once in the test set and $k-1$ times in the training set during the k cross-validation experiments. The results from the k different runs are then averaged. In general, k-fold cross-validation can be repeated several times, each time with a different seed for randomly splitting the dataset. The more k-fold cross-validation is repeated, the lower the variance of the estimates.

The MetaSeer system uses *document-level k-fold cross-validation*, where documents are distributed into k disjoint sets, and then contexts of numbers are extracted in each set. This way, all examples belonging to the same document belong to the same set, training or test.

The average performance after k-fold cross-validation is presented to the user. If the user is not satisfied with the classifier's performance, then the user is asked to manually label more unlabeled data for achieving better accuracy classification. Otherwise, the resulting classifier is used to automatically label large amounts of unlabeled data that are later presented to the user for meta-analysis. The MetaSeer Classification component takes as input the classifier output by the Trainer and an unlabeled dataset of contexts, and it outputs predicted labels for each context corresponding to relevant or non-relevant numbers for the meta-analysis performed by a user.

System Evaluation and Discussion

Dataset

We used a corpus of documents that were compiled for a reliability meta-analysis task, focused on the Organizational Commitment Questionnaire (Porter et al. 1974). The number of documents in our corpus is 100, out of which we ignored scanned documents or documents for which the text was incorrectly extracted. The numbers in this document corpus were manually coded with help from a graduate student majoring in Learning Technologies. Because we are in the proof-of-concept phase of development, we have limited our focus to a small, selected set of variables including sample size, response rate, gender, Likert scale start, Likert scale points, and descriptive statistics associated with the OCQ (mean and standard deviation). In particular, numbers associated with these variables represent the positive examples in our labeled set. In future, we will not only use a larger corpus size, but we will also use other variable types.

MetaSeer Data Extractor Evaluation

We tested the initial version of the MetaSeer Data Extractor (MetaSeerExtractor.1) using the above corpus. By manual inspection of all 100 documents, we found that the extraction process of the MetaSeer Data Extractor was highly accurate. However, numbers expressed in words were not coded originally in our implementation. Through error analysis, our initial testing revealed that such numbers represented as words (e.g., "one hundred and forty-four" as shown in Figure 1) represent numbers of interest, relevant for meta-analysis. We revised our implementation to extract not only numbers (e.g., "118" or "35%" from Figure 1) but also complex numbers expressed as words (e.g., "one hundred and forty-four" in Figure 1). In this version of the MetaSeer Extractor, we ignore numbers from tables. We found that the context of numbers from tables is typically insufficient for the user to comprehend what a number represents. This is a limitation of our current implementation as many data elements that are necessary for a meta-analysis may be documented in tables including but not limited to means, standard deviations, and reliability estimates. We will address this limitation in future work.

In this step, the documents from which we were able to extract the text, were manually coded using the user interface presented in Figure 3. When needed, the annotator used the original pdf documents to determine an accurate annotation. However, we found the annotation process much faster and less error-prone because our system presents to a user only good candidate numbers and avoids the overwhelming irrelevant amounts of information available in documents.

MetaSeer Trainer Evaluation

We tested the reliability of the automatic prediction (or labeling) process by comparing the labels predicted by the trained classifier within the Trainer component with the labels that were previously hand coded. We trained Naïve Bayes classifiers on the "bag of words" representation of contexts (i.e., the contexts' term frequency representations). The results of our experiments are shown in Table 1. We experimented with two settings: one in which the negative examples are taken from the entire document text and another one in which the negative examples are taken only from the "Methods" section. All positive examples are used in both settings.

We reported the following performance measures: accuracy, precision, recall, and f-score. These measures are given as follows: $Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$

Accuracy	Precision	Recall	F-score
With negative instances from the "Methods" section.			
76.34%	0.863	0.763	0.799
With negative instances from the entire text.			
78.69%	0.896	0.786	0.829

Table 1: Evaluation of MetaSeer Trainer in 5-fold cross-validation experiments.

$Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $F\text{-}Measure = \frac{2 \cdot Precision \cdot Recall}{Prec+Recall}$, where TP, FP, FN represent the true positives, false positives, and false negatives, respectively. Given the set of labeled contexts, we can easily compute TP, FP, FN as follows: TP are numbers identified as relevant, which are indeed relevant; FP are numbers identified as relevant, which in fact are not relevant; FN are numbers that are relevant but are not identified by the algorithm as relevant; and TN are numbers identified as non-relevant, which are indeed non-relevant.

As can be seen from the table, our trained classifiers generally achieve high precision. However, the recall is not very high, implying that the number of false negatives is large. The performance is generally lower when negative examples are used only from the "Methods" section.

Conclusion

In this paper, we presented an information technology system called MetaSeer.STEM that can help improve meta-analysis tasks in STEM education research. Our system automatically parses a collection of documents, extracts the candidate elements for meta-analysis, and further classifies them as relevant to a meta-analysis or as non-relevant. The results of our experiments using a "bag of words" representation, in conjunction with Naïve Bayes classifiers, show promising results. In future, it would be interesting to explore NLP techniques to design new features, e.g., based on parse trees. It would also be interesting to evaluate our models on data from different domains. Furthermore, we would like to examine the effect of the window length of contexts around the numbers, especially to estimate how large the context around a number needs to be in order to determine that the number is relevant.

Acknowledgments

We thank our anonymous reviewers for their constructive feedback. This research was supported in part by a seed grant from the University of North Texas.

References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning.* Springer.

Hyde, J.; Lindberg, S.; Linn, M.; Ellis, A.; and Williams, C. 2008. Gender similarities characterize math performance. *Science.* 321:494–495.

Hyde, J.; Fennema, E.; and Lamon, S. 1990. Gender differences in mathematics performance: a meta-analysis. *Psychol Bull.* 107(2):139–55.

Kiritchenko, S.; de Bruijn, B.; Carini, S.; Martin, J.; and Sim, I. 2010. Exact: automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making* 10(56).

Korkontzelos, I.; Mu, T.; Restificar, A.; and Ananiadou, S. 2011. Text mining for efficient search and assisted creation of clinical trials. In *Proceedings of the ACM Fifth International Workshop on Data and Text Mining in Biomedical Informatics*, DTMBIO '11, 43–50.

Lindberg, S.; Hyde, J.; and Hirsch, L. 2008. Gender and mother-child interactions during mathematics homework. *Merrill-Palmer Quarterly.* 54:232–255.

Lindberg, S. M.; Hyde, J. S.; and Petersen, J. L. 2010. New trends in gender and mathematics performance: A meta-analysis. *Psychol. Bull.* 136(6):1123–1135.

Luttman, S.; Mittermaier, L.; and Rebele, J. 2003. The association of career stage and gender with tax accountants? work attitudes and behaviors. *Advances in Taxation* 15:111–143.

McCallum, A., and Nigam, K. 1998. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization, 1998.*

Mitchell, T. M. 1997. *Machine Learning*. McGraw Hill.

Nielsen, F.; Balslev, D.; and Hansen, L. 2005. Mining the posterior cingulate: segregation between memory and pain components. *Neuroimage* 27(3):520–32.

Porter, L.; Steers, R.; Mowday, R.; and Boulian, P. 1974. Organizational commitment, job satisfaction, and turnover among psychiatric technicians. *Journal of Applied Psychology* 59:603–609.

Restificar, A., and Ananiadou, S. 2012. Inferring appropriate eligibility criteria in clinical trial protocols without labeled data. In *Proceedings of the ACM Sixth International Workshop on Data and Text Mining in Biomedical Informatics*, DTMBIO '12, 21–28.

Schmid, R. F.; Bernard, R. M.; Borokhovski, E.; Tamim, R. M.; Abrami, P. C.; Surkes, M. A.; Wade, C. A.; and Woods, J. 2014. The effects of technology use in postsecondary education: A meta-analysis of classroom applications. *Comput. Educ.* 72:271–291.

Schmidt, F. L., and Hunter, J. E. 1977. Development of a general solution to the problem of validity generalization. *Journal of Applied Psychology* 62:529–540.

Schmidt, F. 2008. Meta-analysis: A constantly evolving research integration tool. *Organizational Research Methods* 11:96–113.

Vacha-Haase, T. 1988. Reliability generalization: Exploring variance in measurement error affecting score reliability across studies. *Educational and Psychological Measurement* 58:6–20.

Wu, J.; Williams, K.; Chen, H.-H.; Khabsa, M.; and Caragea, C. 2014. Citeseerx: Ai in a digital library search engine. In *Proceedings of the 26th Annual Conference on Innovative Applications of Artificial Intelligence*.

Yarkoni, T.; Poldrack, R.; Nichols, T.; Van Essen, D.; and Wager, T. 2011. Large-scale automated synthesis of human functional neuroimaging data. *Nat Methods* 8(8):665–70.