Text Categorization for Assessing Multiple Documents Integration, or John Henry Visits a Data Mine

Peter Hastings[1]*, Simon Hughes[1], Joe Magliano[2], Susan Goldman[3], and Kim Lawless[3]

[1] DePaul University
[2] Northern Illinois University
[3] University of Illinois Chicago

Abstract. A critical need for students in the digital age is to learn how to gather, analyze, evaluate, and synthesize complex and sometimes contradictory information across multiple sources and contexts. Yet reading is most often taught with single sources. In this paper, we explore techniques for analyzing student essays to give feedback to teachers on how well their students deal with multiple texts. We compare the performance of a simple regular expression matcher to Latent Semantic Analysis and to Support Vector Machines, a machine learning approach.

Keywords: Natural Language Processing, Machine Learning, Corpus Analysis

1 Introduction

In the digital age, literacy requires the reader more than ever before to be able to gather, analyze, evaluate, and synthesize complex and sometimes contradictory information across multiple sources and contexts [1]. Unfortunately, reading is typically taught and assessed using a single source text and rarely addresses comprehension and learning across multiple sources [2]. To improve this situation, teachers and students must be provided with educational curricula and tools that feature multiple-text comprehension and provide examples of tasks, texts and student performance in different subject matter areas [3–7]. In [2], we described the development of a formative assessment tool for characterizing a student's ability to comprehend and synthesize multiple texts. The goal of the current study is to develop and test techniques for providing automated assessment of the student essays.

As described in [2], 247 middle school students were given three texts describing different factors that led to the population boom in Chicago during the

* The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305F100007 to University of Illinois at Chicago. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

mid-1800s. Each text focused on a different factor that either pushed people from their homes to Chicago (e.g., poor economic opportunities in rural areas), pulled people to Chicago (e.g., increase number of low-skilled jobs, jobs in the railroad industry), or the development of an infrastructure that supported a population increase (e.g., development of railroad and shipping industries). In this paper, these source texts are referred to as the "Better life", "Industry", and "Transportation" texts respectively. Students were told to read the texts and use the content to write an essay explaining why Chicago became a big city.

A critical component of the formative assessment tool is a theoretically-driven, ideal representation of how the texts could be used to answer the question of why Chicago became a big city, called a *documents model* [8, 9]. Created by discourse experts, the documents model is a graph which depicts the cause-and-effect relationships within the set of source texts, as well as the specific details that support these relationships. For example, code CL1 represents the most general level of the pull factors that brought people to Chicago. Code SCL1.1 represents an underlying cause, e.g. "businesses grew." Code SCL1.2 represents the effect of that cause, e.g. "jobs were created." Code ESCL1.2 is a specific example of job creation in meat processing industries. There are 37 codes in the documents model representing the concepts and relationships of the three texts.

In this paper, we describe our efforts to automatically identify the overlap between the student texts and the original source texts. We start by describing the corpus of student texts. Then we present a simple text classification method in which a human expert creates regular expressions to identify student sentences which correspond to a particular documents model category. In section 4, we evaluate Latent Semantic Analysis for classifying the student texts. Then we describe a machine learning approach to the classification problem, and finish with a comparison of the approaches.

2 The Corpus

As described above, our classification task is to determine how student essays relate to the original source texts. Our training data for the different methods was the set of student essays mentioned above that had been coded by human analysts. We worked with 459 student essays collected in 2008 and 2009, consisting of a total of 4076 sentences.

As reported in [2], each student sentence was given a (possibly empty) set of "text codes" that indicated which particular sentence(s) from the three sources it related to. Each sentence was also given a (possibly empty) set of documents model codes which indicated the related concepts from the documents model. For example, the student sentence, "Many people also worked in the meat processing by cutting the cattle and pigs" was coded with text code I16 for sentence 16 of the Industry text: "Butchers cut the cattle and pigs into the meat that people bought in grocery stores." It was coded with documents model code ESCL1.2, described above.

The annotated texts were translated into XML to facilitate the creation of multiple views of the text, for example, sorting by source category, or documents model concept. The sentences were preprocessed by removing punctuation and stop words (using the CLEF english stopword list available from `http://members.unine.ch/jacques.savoy/clef/englishST.txt`) and eliminating words which only occurred in one document. We did not use stemming. All words were upcased.

3 Pattern Matching

Our initial approach to classifying student texts used a tried and true approach: pattern matching with regular expressions. In the spirit of [10], we thought that human ingenuity, combined with a simple technique and a quick and convenient method for refining results might be fruitful. For this analysis, we wanted to determine how well we could identify which student sentences were associated with the codes in the documents model (DM). We created a web-based tool which displayed all the student sentences, sorted by DM code. For each code, it allowed the user to create a regular expression using terms and wildcards. For example, the pattern: `(meat (processing | packaging) * (industry | industries | factories))` matches any sentence that includes the word "meat" followed by "processing" or "packaging" followed by any number of other words and then "industry", "industries", or "factories". The user can submit the set of patterns and receive almost instantaneous feedback about the performance of those patterns in classifying the student sentences in accordance with the human coding.

The concept nodes in the documents model (DM) are arranged hierarchically. The nodes at the top of the hierarchy represent the most general statements about the assigned topic, and therefore can be expressed in great variety of ways. Lower level nodes represent more specific information, which is more likely to be expressed with predictable content words, so we focused our efforts on developing patterns to match these lower level nodes (14 of the 37 total). Table 1 presents the performance of the patterns (and the aggregate) in terms of Recall *(true positives / (true positives + false negatives))*, Precision *(true positives / (true positives + false positives))*, and F_1 score *(2*Precision*Recall/(Precision+Recall))*.

Table 1. Matching documents model codes with regular expressions

DM code	Rec.	Pre.	F_1	DM code	Rec.	Pre.	F_1	DM code	Rec.	Pre.	F_1
ESCL1.1	0.78	0.61	0.68	ESCL2.4	0.78	0.94	0.85	SCL2.1	0.75	0.63	0.68
ESCL1.2	0.73	0.69	0.71	ESCL3.1	0.74	0.56	0.64	SCL2.2	0.92	0.56	0.70
ESCL1.3	0.69	0.80	0.74	ESCL3.2	0.66	0.25	0.36	SCL3.1	0.72	0.74	0.73
ESCL2.1	0.84	0.94	0.89	SCL1.1	0.76	0.86	0.81	SCL3.2	0.62	0.27	0.38
ESCL2.3	0.83	0.93	0.88	SCL1.2	0.78	0.30	0.43	Aggregate	0.76	0.78	0.77

Overall, the performance of this set of patterns was at least respectable and in some cases, very good. Some of the patterns were very simple. For SCL2.2, the pattern was simply a disjunction of the terms, "families", "family", or "feed", and it achieved a very high Recall value. Its Precision was moderate, however, because a significant number of sentences associated with other codes also included these terms. This highlights the difficulty of the "semantic overlap problem". In the case of "hand-built" mechanisms like this one, the problem is especially difficult because there is no way to know if a particular pattern is optimal or how close to optimal it is. For this reason, and to allow a broader coverage of the classification space, we explored automatic methods of classification using Latent Semantic Analysis and Machine Learning.

4 Latent Semantic Analysis

Latent Semantic Analysis (LSA) has been used in a wide range of cognitive modeling and educational tasks [11]. It uses singular value decomposition to create a vector-based representation of the words and documents in the training corpus, and can then compare documents with the cosine measure. Because the nodes in the documents model are conceptual and not textual, we used LSA to compare the sentences of the student essays with the original source sentences (the text model, or TM). This can be used as a proxy for the conceptual analysis, because the documents model includes a mapping from the text model codes to the documents model codes.

We used LSA from `http://lsa.colorado.edu` with the "General Reading up to 1st year college (300 factors)" space to calculate the cosine similarity between each student sentence and each sentence in the three source documents that the students read. If the cosine was greater than a threshold, we assigned the relevant TM code to the student sentence. As with the coder annotations, this allowed multiple TM codes per student sentence. Because the threshold must be empirically derived, we used a range of cosine thresholds (0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, and 0.8). The results are shown in Table 2. The trade-off between Recall and Precision can be clearly seen across the different threshold values. The best result, using F_1 which gives Recall and Precision equal weight, was achieved with a cosine threshold of 0.70.

Table 2. Evaluation of LSA with different cosine thresholds

Threshold	0.40	0.50	0.55	0.60	0.65	0.70	0.75	0.80
Recall	0.70	0.63	0.58	0.53	0.48	0.43	0.38	0.34
Precision	0.14	0.24	0.31	0.41	0.53	0.66	0.75	0.80
F_1	0.23	0.35	0.40	0.46	0.50	0.52	0.50	0.48

5 Machine Learning

In a machine learning approach to text classification, some set of features of the texts are used to induce a classifier that should correctly categorize as many of the texts as possible. The most obvious features of a document are the words within it. One popular learning method for this type of classification task is Support Vector Machines (SVMs) [12, 13]. In this section we describe other applications of text classification techniques in educational contexts and then present our approach and evaluations of it.

5.1 Related Work

Although there have been a great many applications of machine learning in text classification for information retrieval, there have been relatively few within an educational context, and most of them have been aimed at inferring dialog acts, for example [14]. More similar approaches to ours include Larkey's [15] comparison of k-nearest neighbor, naïve Bayes and linear regression classifiers in assigning grades to student essays. Sathiyamurthy and Geetha [16] built a text classification system which used part-of-speech tagging to align e-learning documents according to an ACM domain ontology, allowing the documents to be classified according to Bloom's taxonomy [17]. Yilmazel et al [18] used an SVM algorithm to perform text categorization for automatically aligning curricular documents with state and federal science benchmarks.

5.2 SVMs for text classification

A typical task for text classification is learning to categorize news articles by topic. In [12], for example, SVMs were trained to identify the topic of 800,000 news stories from Reuters at three different levels of granularity. Two important differences between that study and ours are the size of the individual documents and the size of the training set. Because we would ideally like to give teachers information about which concepts from the documents model are included in the student essays, we are most interested in classifying individual sentences (as opposed to paragraph-length or longer documents). As mentioned above, our entire corpus consists of approximately 4000 sentences, two orders of magnitude less than Medlock used.

To create the training data for the SVMs, we separated the student essays into sentences (= documents) and preprocessed them as described above (removing stop words, etc.). Then we computed normalized *tfidf* vectors for each document following [13]. Each document vector had a weight for each of the terms in it. The weight for a term was computed as the number of times it occurs in the document divided by the log of the number of documents it occurs in. Then each vector is normalized to have length = 1 to allow comparison of documents with differing numbers of terms.

We used 10-fold cross-validation along with svm_multiclass [13] and trained the classifiers to categorize the sentences into the 37 documents model categories. The results from the best performing model are shown in Table 3. For comparison, the 14 DM codes which were also included in the pattern matching evaluation are shown in italics. The penultimate entry is the aggregate across all categories. The row labelled "Aggr 14" shows the aggregate results across the 14 codes from the pattern matching evaluation.

Table 3. SVM performance for DM codes

DM code	Rec.	Pre.	F_1	DM code	Rec.	Pre.	F_1	DM code	Rec.	Pre.	F_1
A	0.59	0.42	0.49	ESCL3	0.00	0.00	0.00	RC3	0.05	0.03	0.04
CL1	0.20	0.31	0.24	*ESCL3.1*	0.63	0.44	0.52	RC3.1	0.10	0.25	0.14
CL2	0.49	0.40	0.44	*ESCL3.2*	0.58	0.47	0.52	RC3.2	0.00	0.00	0.00
CL3	0.44	0.44	0.44	IRC1	0.02	0.10	0.03	RC3.3	0.08	0.19	0.11
ESCL1	0.59	0.45	0.51	IREN1	0.00	0.00	0.00	RE1	0.00	0.00	0.00
ESCL1.1	0.80	0.51	0.62	IREN2	0.00	0.00	0.00	*SCL1.1*	0.29	0.33	0.31
ESCL1.2	0.85	0.52	0.65	RC1+2	0.00	0.00	0.00	*SCL1.2*	0.46	0.46	0.46
ESCL1.3	0.87	0.65	0.74	RC1.1	0.08	0.26	0.12	*SCL2.1*	0.20	0.28	0.23
ESCL2	0.04	0.14	0.06	RC1.2	0.00	0.00	0.00	*SCL2.2*	0.24	0.27	0.25
ESCL2.1	0.63	0.43	0.51	RC2.1	0.06	0.14	0.08	*SCL3.1*	0.16	0.27	0.20
ESCL2.2	0.60	0.51	0.55	RC2.2	0.07	0.20	0.10	*SCL3.2*	0.02	0.25	0.04
ESCL2.3	0.72	0.49	0.58	RC2.3	0.06	0.17	0.09	Aggregate	0.42	0.42	0.42
ESCL2.4	0.70	0.49	0.58	RC2.3A	0.01	0.17	0.02	*Aggr 14*	0.54	0.45	0.49

Of the codes that were matched with the regular expression approach, the SVM often achieved better Recall but worse Precision. As mentioned above, "casting a broader net" increases Recall, but reduces Precision. Overall, these results confirm our intuition that the more specific concepts would be the easier ones to match. The exception to this is the A code (for Assertion). This is a sort of "catch-all" category that indicates a factual statement made by the student which is not directly derived from any of the sources. Despite the breadth of this category, the SVM achieved respectable performance in identifying it. It must be mentioned, however, that the A code is the most frequent one in the corpus, assigned to almost 1300 sentences, 18% of the total of 7321 TM codes given by the human coders. This compares with an average of 191 sentences (2.6%) for the codes in the subset of 14 used in pattern matching. Thus, it is possible, and perhaps even likely, that the SVM's performance on those more specific categories suffered for the benefit of overall performance. This will be discussed further in the next section.

6 Discussion, Future Work, and Conclusions

As shown above, among the 14 DM codes that pattern matching was applied to, the SVM approach significantly outperformed the pattern matching approach in only one of the categories, ESCL3.2. For the rest, pattern matching was close or much better. When training the SVM, we noticed that with tighter margins between the learned set of support vectors and the training set (lower values of the C parameter), prediction of many of the semantic categories was good, except for the catch-all A category. Because it is the most frequent, that had a large effect on the overall performance. By increasing the margin, we were able to improve performance on A and overall, but with reduced performance on the categories which had fewer examples in the training sets. However, we also tried creating binary classifiers for each DM code (not reported here due to space limitations). This would at least partially address the concern about the relative frequencies of the categories. Each binary classifier only has to distinguish the members vs. non-members of one category. There is still an effect, however, of the small number of positive instances of the more specific categories relative to the entire training set. The binary classifiers that we created generally achieved good Recall but poor Precision.

If we use the entire set of categories, we can (almost) directly compare the three approaches, but pattern matching gets a much lower Recall score (0.18) due to the missing codes. In this comparison, $F_1(Patterns) = 0.29$, $F_1(LSA) = 0.52$, and $F_1(SVD) = 0.42$. Although LSA matched student sentences with TM codes instead of DM codes, the aggregate measure should provide a good idea of the overall performance. We suspect that LSA had an advantage over SVM because many of the student sentences were close paraphrases of the source sentences. We should be able to check this by inferring DM codes from TM codes. This will be done in future work.

One advantage that the pattern matching approach has over the others is that it can take the ordering of the words into account. This could be addressed in a machine learning context by using n-grams or term identification methods. In future work, we will also explore other variations of the machine learning methods, including different classification techniques and higher-level approaches like boosting. If pattern matching retains its advantage for particular codes, a hybrid approach can be developed.

In this paper, we explored three text classification methods, pattern matching, LSA, and SVMs. For identifying many of the specific semantic categories, pattern matching performance exceeded that of the automatic methods. Despite the limitations of the pattern matching approach — the difficulty of coming up with appropriate patterns for all of the categories and the impossibility of knowing what an optimal pattern is — we believe that such a simple technique can still be effective and useful in an educational context.

References

1. New London Group: A pedagogy of multiliteracies: Designing social futures. Har-

vard Educational Review **66** (1996) 60–92
2. Goldman, S.R., Lawless, K.A., Gomez, K.W., Braasch, J.L.G., MacLeod, S., Manning, F.: Literacy in the digital world: Comprehending and learning from multiple sources. In McKeown, M.G., Kucan, L., eds.: Bringing Reading Researchers to Life. Guilford, NY (2010) 257–284
3. Britt, M.A., Wiemer-Hastings, P., Larson, A., Perfetti, C.: Using intelligent feedback to improve sourcing and integration in students' essays. International Journal of Artificial Intelligence in Education **14** (2004) 359–374
4. Britt, M.A., Kurby, C., Dandotkar, S., Wolfe, C.: I agreed with what? Memory for simple argument claims. Discourse Processes **45**(1) (2008) 52–84
5. Goldman, S.R., Bloome, D.M.: Learning to construct and integrate. In Healy, A.F., ed.: Experimental Cognitive Psychology and its Applications: Festshrift in Honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer. American Psychological Association, Washington, D.C. (2005) 169–182
6. Wolfe, M.B., Goldman, S.R.: Relationships between adolescents' text processing and reasoning. Cognition & Instruction **23**(4) (2005) 467–502
7. VanSledright, B.: Confronting history's interpretive paradox while teaching fifth graders to investigate the past. American Educational Research Journal **39** (2002) 1089–1115
8. Rouet, J.F.: The skills of document use. Erlbaum, Mahwah, NJ (2006)
9. Rouet, J.F., Britt, M.A.: Relevance processes in multiple document comprehension. In McCrudden, M.T., Magliano, J.P., Schraw, G., eds.: Text Relevance and Learning from Text. Information Age Publishing, Greenwich, CT (in press)
10. Hobbs, J., Appelt, D., Tyson, M., Bear, J., Israel, D.: SRI International: Description of the FASTUS system used for MUC-4. In: Proceedings of the Fourth Message Understanding Conference, San Mateo, CA, Morgan Kaufmann Publishers, Inc. (1992)
11. Landauer, T., Dumais, S.: A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. Psychological Review **104** (1997) 211–240
12. Medlock, B.: Investigating classification for natural language processing tasks. PhD thesis, University of Cambridge (2007) Technical Report UCAM-CL-TR-721.
13. Joachims, T.: Learning to Classify Text Using Support Vector Machines. PhD thesis, Cornell University (2002) Kluwer.
14. Samuel, K., Carberry, S., Vijay-Shanker, K.: Computing dialogue acts from features with transformation-based learning. In: Papers from the 1998 AAAI Spring Symposium on Applying Machine Learning to Discourse Processing. Number SS-98-01, Menlo Park, CA, AAAI Press (1998) 90–97
15. Larkey, L.S.: Automatic essay grading using text categorization techniques. In: Proceedings of SIGIR'98. (1998) 90–95
16. K.Sathiyamurthy, T.V.Geetha: Association of domain concepts with educational objectives for e-learning. In: Proceedings of Compute'10. (2010) 330–333
17. Bloom, B., ed.: Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. Longmans, Green, New York (1956)
18. Yilmazel, O., Balasubramanian, N., Harwell, S.C., Bailey, J., Diekema, A.R., Liddy, E.D.: Text categorization for aligning educational standards. In: Proceedings of the 40th Hawaii International Conference on System Sciences. (2007) 73–80