Automated approaches for detecting integration in student essays

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

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

Abstract. Integrating information across multiple sources is an important literacy skill, yet there has been little research into automated methods for measuring integration in written text. This study investigated the efficacy of three different algorithms at classifying student essays according to an expert model of the essay topic which categorized statements by argument function, including claims and integration. A novel classification algorithm is presented which uses multi-word regular expressions. Its performance is compared to that of Latent Semantic Analysis and several variants of the Support Vector Machine algorithm at the same classification task. One variant of the SVM approach worked best overall, but another proved more successful at detecting integration within and across texts. This research has important implications for systems that can gauge the level of integration in written essays.

Keywords: support vector machines, latent semantic analysis, multi-word regular expressions, integration, document classification

1 Introduction

Researchers and teachers have recognized that a fundamental challenge for education is teaching students to be able to read with *deep understanding.* To thrive in society students need to learn how to select and evaluate multiple sources of information, make connections across sources (even when information is contradictory) and to apply what they discover to achieve their goals. These critical skills of reasoning within and across texts have been included in the U.S. Common Core Standards of education (http://www.corestandards.org/in-the-states).

Methods for teaching these skills will require the use of open-ended tasks like writing integrative essays. Previous work has explored the use of automated

* The project described in this article is funded, in part, by the Institute for Education Sciences, U.S. Department of Education (Grant R305G050091 and Grant R305F100007). The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept of Education. Correspondence may be sent to Peter Hastings, DePaul Univ. CDM, 243 S. Wabash, Chicago IL 60604, USA.

analysis of essays [1, for example], but mostly this has focused on summaries or analyses of single texts. Our overall goal is to teach students how to read and understand texts more deeply by having them write summaries that combine ideas from multiple texts. This poses two major challenges for automated essay analysis. The first is the semantic overlap of the texts. Some amount of overlap is necessary to help them make inferences. Yet this is problematic for automated techniques, particularly those that rely on word occurrences rather than text structure. The second challenge is cross-text inferences. Although a key goal of this project is to teach students to make such inferences, the broad variety of connections that students construct can make them harder to detect by automatic mechanisms. This paper analyses three different mechanisms that can be used as the evaluation component in a system to assist students to learn to integrate material across texts.

2 Document Classification Algorithms

2.1 Latent Semantic Analysis

Document classification techniques fall into two areas, 'bag-of-words' approaches which ignore word order, and order sensitive methods. This study investigates two bag of word approaches, Latent Semantic Analysis (LSA) and Support Vector Machines (SVM's), and multi-word, which is order sensitive. LSA was initially developed as an Information Retrieval system, but was later found to closely model human lexical acquisition in a number of ways [8]. It creates a term-document co-occurrence matrix where each cell is weighted for the frequency of the term in the document relative to the entire corpus [6]. Then singular value decomposition re-orients the data axes, ranked by their correlation with the data. The top K dimensions (typically 300–400) are used to compare texts. LSA has been used for text classification in applications such as comparing student answers to expected answers in an ITS [3] and grading student essays [2, for example]. For text classification, a threshold cosine value is chosen to achieve the best correlation with human similarity judgments.

2.2 Support Vector Machines

SVM's were introduced as a binary classifier for classifying non-linearly separable classes. An SVM creates one or more hyperplanes in higher-dimensional space that allow linear separation of the data points into separate classes by selecting the hyperplane with the largest margin of separation. This minimizes generalization error. Multiclass SVM's have subsequently been developed [5]. SVM's have been successful at tackling a wide range of regression and classification problems, including text classification [11, for example]. Several authors have tried to improve SVM classification performance by combining them with techniques that take into account word order, with mixed results [11, for example].

2.3 Multi-Word

Ignoring word order when classifying text ignores useful semantic information, motivating research into the multi-word approach. There are 2 main variants, a syntactic and an n-gram approach. The syntactic technique extracts re-occurring phrases consisting only of nouns, adjectives and propositions that follow a particular syntactic structure [7, 11]. The n-gram approach looks for the occurrence of any n-word phrase with a frequency above a threshold [10]. The extracted phrases are then typically used as features for some other classification approach, such as an SVM [11], or to enhance queries used to classify documents [10]. The approach has proven successful in a number of empirical studies [10, 11] .

3 Methodology

3.1 Data

In 2008 and 2009, students from grades 5–8 in two large urban public schools were asked to read three short articles (around 30 sentences each) about Chicago history, and then write essays about population growth in Chicago. 365 essays were written. The articles were created to be complementary, with minimal semantic overlap. One article covered "push" factors driving people to the city, another detailed "pull" factors pulling people to Chicago. The third described how advances in transportation enabled this migration. An *integrated model* was created to represent the conceptual content of the articles and likely connections that students might make between and within the articles, and between the articles and the overall question about population growth in Chicago. The conceptual content was hierarchically structured in the model into high-level claims, intermediate evidence supporting the claims, and low-level details about the evidence. Human annotators coded the correspondence between the student sentences and both the sentences of the articles and the (37) nodes of the integrated model. The inter-rater reliability for the two coders was 85%.

3.2 Metrics

Three metrics were used to measure classification performance across the different approaches, recall, precision and F_1 score, as described in [6, p. 578]. Recall measures false negatives, and thus Type II errors, while precision measures the number of false positives, and thus Type I errors. Typically, as recall increases, precision decreases and vice versa. A combined measure is commonly used to evaluate performance, the F measure, using a coefficient β to adjust the weighting of recall to precision [6, p. 578]. To evaluate the classification performance of each approach, we performed ten-fold cross-validation [9, p. 112].

3.3 LSA

We previously used LSA to identify how many sources the students were referring to [4]. We used the lsa.colorado.edu site to compare student sentences with the

sentences of the articles. A more important goal was to determine how well the students covered the concepts in the integrated model. To do this, we used the correspondences which were specified in the model between the nodes and the article sentences. Many of the nodes had multiple associated sentences, and many sentences had multiple associated nodes. The 7 "linking" nodes reflected an inference between part of an article to part of another article, or to the overall claim of the essay, and so had no corresponding article sentences. If the LSA cosine between a student's sentence and an article sentence was above a threshold that we determined, the sentence was assigned the code(s) of the model node(s) associated with that sentence. We tested thresholds from 0.4 to 0.8, by 0.05 increments, and found a value of 0.7 had the highest overall F_1 score.

3.4 SVM

In prior work, we compared the performance of an SVM to a manual pattern matcher and LSA, and found that the patterns outperformed the SVM [4]. In that study, we used the multiclass SVM to choose the single most likely class for each test example. But many of our example sentences were assigned multiple codes resulting in these sentences appearing in the dataset multiple times, once for each code. This meant that at most one of these multi-code sentences could be coded correctly by the SVM, thereby limiting the overall performance.

In the current study, we evaluated two methods to overcome this. First we used an SVM binary classifier. For each of the 37 classes, a sentence was marked as a positive instance only if that class was in the set of codes assigned by the human raters. The sentences were represented by a $tfidf$ weighted vector, as with LSA. We trained a different classifier for each code. The second SVM approach used the multiclass method, but in a different way. As well as the "best" prediction for each example, SVMlight gives a weight for each class. We established a threshold, and used it to assign (potentially) multiple classes to each example. To avoid bias in the choice of threshold, we calculated the average number of codes per sentence, then selected a threshold which would produce the same number. The threshold also depends on the C parameter (margin) that the SVM model was trained with, so we repeated the process for a range of C values. The best performance was achieved with a C value of 1000 and a threshold of 0.19. This method is marked as "SVM threshold" in the results.

3.5 Multiword

The multi-word approach used is closest to the n-gram approach and is a binary classifier. The algorithm extracts re-occurring expressions (one or more words long), as described in [10], and iteratively constructs a regular expression to classify each category. For each category, all multi-word phrases were extracted and converted into regular expressions. The category's F_β score was then computed for each expression. The expression with the highest F_β score was removed along with all sentences matching the expression. This process was then repeated, and a composite regular expression was built iteratively by combining the highest

scoring expressions using the 'or' operator. Its classification performance was measured on the validation dataset after each iteration. The algorithm halted either when no expressions remained, or after ten consecutive iterations without improvement on the validation dataset to prevent over-fitting [9, p. 116]. β values of 0.25, 0.5, 1 and 2, were used with 0.25 producing the highest F_1 score.

4 Results and Discussion

Our main goal was to evaluate different methods of detecting integration between sources in sentences making up essays. To do this, the 37 model categories were separated into 5 groups corresponding to higher-level categories in the model, including 2 groups containing sentences showing integration between different texts (IR) and within the same text (RC). The IR category also contains inferred relations between a text and a top-level assertion. The other 3 categories consisted of sentences making top-level claims (CL), evidence for those claims (EV) and details surrounding the evidence (DET). Separating the integration categories from the other categories allows a direct evaluation of the techniques at detecting integration, and the other categorical groupings. These results along with the aggregate classification performance are shown in Figure 1 below.

The SVM binary classifier out-performed the other approaches overall and in the CL, EV, and DET categories, while the SVM threshold method demonstrated the best classification performance on the RC and IR categories and thus was the best approach for detecting integration. These 2 techniques showed significant improvement over the SVM multiclass. The multi-word method had the second highest classification performance on the IR category, although it did poorly on all other categories. LSA performed particularly poorly on the IR category. The LSA approach we adopted classified sentences based on their similarity to individual sentences in the source texts, and thus would perform poorly identifying sentences composed from multiple source sentences. The smallest category (IR) was a challenge for all of the algorithms. Machine learning algorithms often struggle with small datasets [9], which may explain this observation.

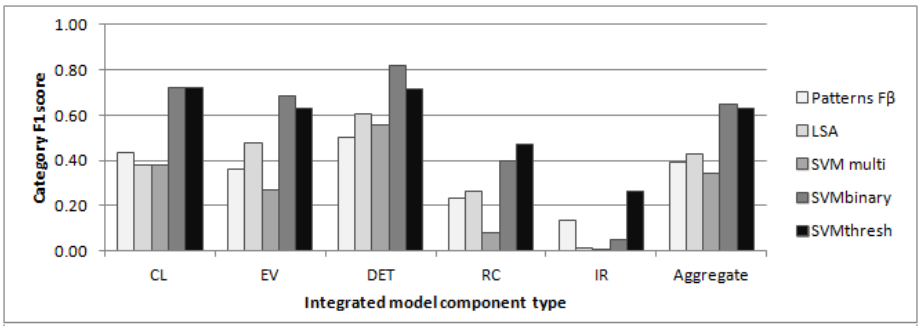

Fig. 1. Aggregate F_1 score by algorithm across different integration model categories

Although SVM's performed better overall, the strong performance of multi-word on the IR category indicates that a hybrid approach combining the threshold SVM method with multi-word may improve the performance at detecting integration. Several authors have used the multi-word approach to create features that were then used by an SVM for text classification [11, for example]. Such an approach may prove more successful at this task than either approach in isolation. Naive Bayes has been successfully applied to text classification and may also be effective at this task. Also, repeating the experiments with a larger dataset with more sentences in the IR category may yield better results. One limitation to this study was the need to create an integrated model of the topic, and manually code a datset to this model. For this approach to be successfully applied to new domains, the manual effort required would need to be minimized. If multiple datasets on different topics were collected, each with their own integrated model, it may be possible to train a more general classifier that can detect integration in unseen datasets without the need for an integrated model.

References

1. Attali, Y., Burstein, J.: Automated essay scoring with e-rater R V.2. Journal of Technology, Learning and Assessment 4, 1–30 (2006)
2. Foltz, P., Britt, M., Perfetti, C.: Reasoning from multiple texts: An automatic analysis of readers' situation models. In: Proceedings of the 18th Annual Conference of the Cognitive Science Society. pp. 110–115. Erlbaum, Mahwah, NJ (1996)
3. Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., the Tutoring Research Group: Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. Interactive Learning Environments 8(2), 129–147 (2000)
4. Hastings, P., Hughes, S., Magliano, J., Goldman, S., Lawless, K.: Text categorization for assessing multiple documents integration, or John Henry visits a data mine. In: Biswas, G., Bull, S. (eds.) Proceedings of the 15th AIED Conference (2011)
5. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods: Support Vector Learning. MIT Press (1999)
6. Jurafsky, D., Martin, J.: Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, New York (2000)
7. Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1(01) (1995)
8. 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, 211–240 (1997)
9. Mitchell, T.: Machine Learning (Mcgraw-Hill International Edit). McGraw-Hill Education (ISE Editions), 1st edn. (Oct 1997)
10. Papka, R., Allan, J.: Document classification using multiword features. In: Proceedings of the seventh international conference on Information and knowledge management. pp. 124–131. CIKM '98, ACM, New York, NY, USA (1998)
11. Zhang, W., Yoshida, T., Tang, X.: A comparative study of tf*idf, lsi and multi-words for text classification. Expert Systems with Applications 38(3) (2011)