# Identifying Content Patterns in Peer Reviews Using Graph-Based Cohesion

**Lakshmi Ramachandran**
Pearson Knowledge Technologies
lakshmi.ramachandran@pearson.com

**Edward F. Gehringer**
North Carolina State University
efg@ncsu.edu

## Abstract

Peer-reviewing allows students to think critically about a subject and also learn from their classmates' work. Students can learn to write effective reviews if they are provided feedback on the quality of their reviews. A review may contain *summative* or *advisory* content, or may *identify problems* in the author's work. Reviewers can be helped to improve their feedback by receiving automated content-based feedback on the helpfulness of their reviews. In this paper we propose a cohesion-based technique to identify patterns that are representative of a review's content type. We evaluate our pattern-based content identification approach on data from two peer-reviewing systems—Expertiza and SWoRD. Our approach achieves an accuracy of 67.07% and an $f$-measure of 0.67.

## Introduction

Collaborative learning systems such as SWoRD (Scaffolded Writing and Rewriting in the Discipline) (Nelson and Schunn 2009) and Expertiza (Gehringer 2010) provide an environment for students to interact with other students, exchange ideas, provide feedback and use peers' text-based reviews to identify mistakes in their own work, and learn possible ways to improve it. The past few years have witnessed a growth in Massive Open Online Courses (MOOCs) such as Coursera and Udacity, as platforms for web-based collaborative learning. MOOCs require a scalable means of assessment, and for papers that are not well-structured or contain figurative information, i.e., cases where Automated Essay Scoring may not work well, peer review fills the bill (Balfour 2013). However, students, who may not have any experience in peer review, need to be guided through the reviewing process. We have developed an automated review analysis application to help keep reviewers in check and improve the quality of peer feedback.

A review may provide an assessment of the kind of work that was done—praising the submission's positive points, identifying problems, if any, and offering suggestions to help improve the submission. A review may contain the following types of content. **Summative:** Positive feedback or a summary of the author's work. E.g. "The page is organized logically, and gives an example code." **Problem-detection:** Identifies problems in the author's submission. E.g. "The page lacks a qualitative approach. It also lacks an overview." **Advisory:** Provides suggestions to the authors on ways to improve their work. E.g. "The page could contain more ethics related links and more in-depth analysis of ethical issues."



These content categories are selected based on empirical studies conducted by Nelson et al. (2009) and Goldin et al. (2010). Nelson et al. found that reviews that locate problems in the author's work or provide suggestions for improvement, helped authors understand and use feedback effectively. Goldin et al. found that the use of problem-specific prompts to support reviewers resulted in more informative (or less redundant) feedback. This type of content categorization was also used by Cho (2008) in his work on identifying the content type of peer reviews in Physics classes.

In a study that surveyed 24 participants on their perceived usefulness of review quality metrics, 17 out of the 24 participants found review content type to be a very useful feedback metric (Ramachandran and Gehringer 2013b). We choose to study content type of reviews in this work since our aim is to use artificial intelligence (AI) and natural language processing techniques to identify and provide feedback to student reviewers on a metric deemed to be useful.

Some of the current approaches to automatically identify review content use machine-learning techniques with shallow text features such as counts of nouns and verbs (Cho 2008). From the above examples of summative, problem-detection and advisory content, we see that they discuss *similar points* (e.g. page organization), but the difference lies in the *way* the points are discussed. For example, summative reviews make positive observations (e.g. "... organized logically..."), while problem-detection reviews identify problems (e.g. "... lacks ... approach ...") and advisory reviews provide suggestions (e.g. "... more ... analysis ..."). Techniques that rely only on token frequencies may not succeed in distinguishing content types containing overlapping text.

In this paper we introduce the important task of identifying the type of content a review contains. Some of the important contributions of this work to the field of AI and education are:

1. Use of a cohesion-based technique to extract semantic patterns from reviews represented as word-order graphs.
2. Use of a lexico-semantic matching that captures the relat-

edness between tokens or phrases.

3. Evaluation of our approach on two state-of-the-art collaborative learning applications: Expertiza and SWoRD.

## Related Work

The assessment of reviews is an important problem in education, as well as science and human resources, and is therefore worthy of serious attention. Cho (2008) uses $n$-grams as features with techniques such as naïve Bayes, support vector machines (SVM) and decision trees to classify reviews as praise, criticism, problem detection or suggestion. Xiong et al.'s (2010) approach uses features such as counts of nouns and verbs to locate problematic instances in a review.

Peer reviews contain opinions and tonal information. However, our aim is not to identify the tonality or sentiments (positive, negative or neutral) expressed in reviews, but to be able to determine their content type–presence of summary, praise, problematic or advisory instances in a review. Tone information and presence or absence of negations help guide the content identification process.

Graph-based approaches have been used for a wide variety of tasks such as text summarization and topic identification. Mihalcea (2004) uses a graph representation to perform sentence extraction. Radev et al.'s (2004) MEAD uses a centroid-based summarization technique to identify the best sentences to be included in a summary. Erkan et al. (2004) use a centrality-based technique to determine the main ideas in a document. Coursey et al. (2009) determine the topic of an input document by identifying the central vertex using Google PageRank. Our approach selects the most similar graph edges to represent a content type's patterns.

## Motivating Example

The review in Figure 1 is written for an article on *software extensibility*. The sample review's content type is rated on a scale of 0–1. Some of the patterns identified in this review are highlighted. The highlighted patterns and numeric estimates give the authors information on the types of content their review contains. A low numeric score means that the reviewer should add more content of that particular type.

The categories discussed in this paper are used only to identify the *content type* of reviews. There are several other dimensions based on which a review's quality can be judged. For instance (1) determining whether a review's content is relevant to the author's submission (Ramachandran and Gehringer 2013a), (2) identifying whether a review covers the main topics discussed by the author, without digressing or focusing on just one tiny part of the author's work (Ramachandran, Ravindran, and Gehringer 2013), (3) determining the tone or semantic orientation of a review. As can be seen from the screenshot in Figure 1, our automated review analysis system provides formative feedback on some of these other metrics. These other metrics are complex and detailed descriptions of the approaches used to compute them are beyond the scope of this paper.

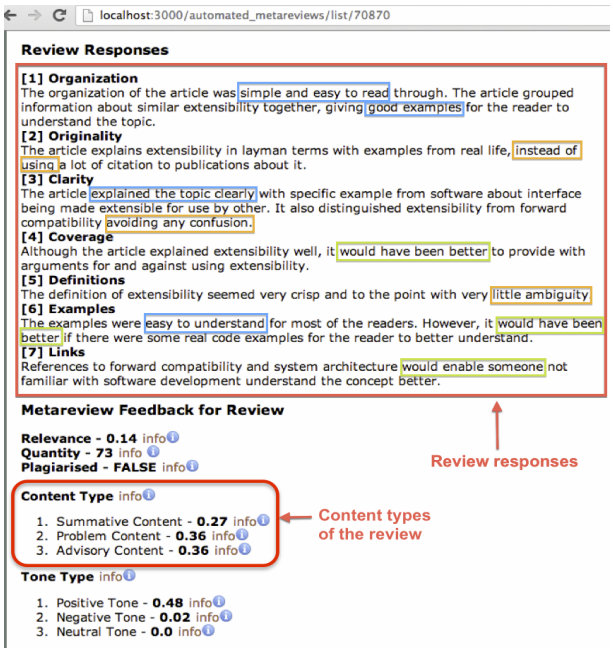


Figure 1: Output from our review assessment system identifying content type of a review. Summative patterns are highlighted in *blue*, problem-detection patterns are highlighted in *orange* and advisory patterns are highlighted in *green*.

## Approach

### Generating Word Order Graphs

We use word-order graphs to represent text. Word-order graphs contain the ordering of words or phrases in a text and help capture context information. Context is not available in a bag-of-words or a dependency tree type representation (which captures only head → modifier information). Ramachandran et al. (2012) have shown that context-based word-order graphs are useful for the task of identifying a review's relevance to a submission (the text under review).

During graph generation, each review is tagged with parts-of-speech (POS) using the Stanford POS tagger (Toutanova et al. 2003) to help identify nouns, verbs, adjectives, adverbs etc. in a review. Graph vertices may contain tokens or phrases. Vertices are connected by edges. Graph edges represent dependencies such as subject-verb (SUBJ), verb-object (OBJ) or noun-modifier (NMOD). Dependencies help describe the relation between tokens and their parents. We use the *anna* library in the *mate tools* package to generate dependency tags (Bohnet 2010). Edge labels capture grammatical information, which would be useful during edge matching for pattern identification. A detailed description of the process of generating word-order graphs is available in Ramachandran et al. (2012).

*State* of a sentence (described in the next section) helps determine whether a token or a phrase in a review is being used in a positive, negative or advisory sense. State is identified during graph generation, and is represented as part of a graph's vertex. Figure 2 contains the graph for a sample review. Vertices contain POS of the token e.g. noun, verb or

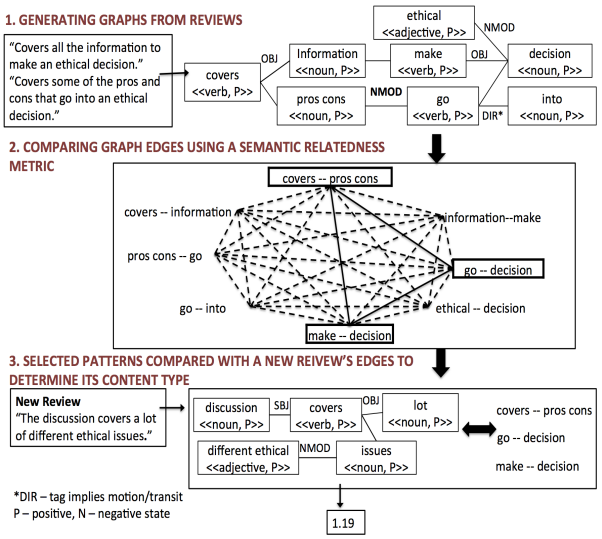


Figure 2: Illustration of our approach–Steps 1 and 2: Patterns are identified from sample summative reviews "Covers all the information to make an ethical decision." and "Covers some of the pros and cons that go into an ethical decision." Step 3: A new review's semantic similarity is identified by comparing it with the generated patterns.

adjective, as well as state information, where $P$ represents positive and $N$ represents a negative use of the token.

**Review state** During semantic matching we look for cases of negation in the review. Consider the review, "The paper is not clear." An approach that does not handle negation is likely to misclassify this review as summative. State of a sentence may be—*positive, negative* or *advisory*. Words such as {*none, never, not, won't, don't, didn't, barely, hardly*} give the text a negative orientation. Tokens such as {*could, should, maybe, perhaps*} are indicators of suggestion.

Chapman et al. (2001) use regular expressions to determine negations of clinical terms in patients' medical records. Manshadi et al. (2013) use negations to determine the scope of ambiguous quantifiers in a text. The state of a review may not be evident when the tokens are looked at independently. We therefore apply a heuristic, rule-based approach to identify state based on tokens and their contexts. Our approach not only identifies negations, but also identifies advisory terms or phrases in reviews.

Reviews are broken down into segments at connectives such as "and", "but" in order to distinguish between the state of each segment. A segment is assigned a default state until a token or phrase of negative or advisory state is identified.

Our approach takes cases of double negatives into consideration, as shown by Harabagiu et al. (2006). We use the presence or absence of nouns in between tokens (context) to determine how double negations should be resolved. For instance, in the text "It is hardly understandable, and the text is incomplete.", the presence of the noun "text" in between "hardly" and "incomplete" causes the state to remain negative. Negative words, separated by tokens, embellish the negative polarity of the text (Tubau 2008). Negations such as "no", "none" and "never" in front of other negative words also strengthen the negation, e.g. "No the explanation does not help!" Consider the segment "It is hardly incomplete." There are no nouns or verbs between the negative descriptors "hardly" and "incomplete". The two negative words cancel each other out, resulting in a positive state.

In the case of advisory indicators, context plays an important role in determining state change. Advisory tokens when followed by a negative token results in a change of state from advisory to negative. In the example "...could not understand...", since the advisory token "could" is followed by "not", the segment gets a negative orientation. However, presence of nouns or verbs between advisory and negative tokens would cause the state to remain advisory. In the case of segment, "I *would suggest* the author to *not* include...", the presence of the noun "author" between the advisory token "would" and the negation "not" causes the sentence to remain a suggestion–advising the author against doing something.

After parsing every token in the segment, the algorithm returns the final state. If no negative or advisory token is identified, the review has a positive state. We manually collected a set of negative indicator words and phrases, found commonly among educational reviews (e.g. "grammatical errors", "off topic", "too short"), from 100 reviews completed using Expertiza (Gehringer 2010). We use additional negative indicators from an opinion lexicon provided by Liu et al. (2005).

## Determining Semantic Similarity Across Edges

We use WordNet (Fellbaum 1998) to determine match between the graph edges. Relatedness is measured as the average of the matches between vertices of two edges that are being compared. Match between two tokens could be one of: (1) exact, (2) synonym, (3) hypernym or hyponym (more generic or specific), (4) meronym or holonym (sub-part or whole) (5) presence of common parents (excluding generic parents such as *object*, *entity*), (6) overlaps across definitions or examples of tokens i.e., using context to match tokens, or (7) distinct or non-match. The seven types of matches are weighted on a scale of 0 to 6. An exact match gets the highest weight of 6, a synonym match gets a weight of 5 and so on, and a distinct or non-match gets the least weight of 0.

When edges are compared, their respective states are compared. If two vertices have the same state then similarity is $+$value, whereas if they have different states, then the similarity is $-$value. For example, if two tokens have an exact match but have different states then they get a match value of $-6$.

## Selecting Edge Patterns

Graph edges from reviews that contain predominantly one type of content are compared with each other (Step 2 in Figure 2). The importance of an edge $e$ is calculated by taking the average of the matches that $e$ has with each of the other edges in the set. Importance is given by the formula in Equation 1, where $E$ is the set of all edges in the graph. Edges such as noun–adjective, verb–adverb capture properties of nouns and verbs in a text and these edges help distinguish

Table 1: Sample edge patterns for each review content type.

| summative | problem detection | advisory |
|---|---|---|
| page–discussed | not–covered | could be–bit |
| parts–original | typing–mistake for | would benefit–more |
| good–examples | grammatical–problems | more–detail |

the *way* in which objects or concepts are discussed by different reviews.

$$\textit{Importance of } e = \frac{1}{|E|-1}\Big(\sum_{\forall f \in E, f \neq e} \textit{similarity}(e,f)\Big) \quad (1)$$

From among the edges generated for reviews of the same content type those that have a high average similarity with other edges are selected as patterns. We select the top 50 patterns from each content type to ensure that the same number of patterns is selected for every content type. Table 1 lists some edge patterns selected from each content class.

## Identifying Content Type of A New Review

Content type of a new review is identified by comparing the edges of the new review's graph with each content type's patterns. The best semantic match for each review edge with a content's patterns is identified. The average of the review edges' matches gives the semantic match between a review and the content's patterns (Equation 2).

$$\textit{content}_C = \frac{1}{|E|}\sum_{\forall e \in E}\Big(\underset{\forall p \in P_C}{\operatorname{argmax}}\ \textit{similarity}(e,p)\Big) \quad (2)$$

In Equation 2, $\textit{content}_C$ represents the degree of match between the new review (with edges $E$) and patterns of content type $C$ ($P_C$), where $C$ could be *summative, problem detection* or *advisory*. Patterns are given state values in order to aid matching. Summative patterns have a positive state, problem-detection patterns have a negative state, and advisory patterns are assigned an advisory state.

In Step 3 in Figure 2 summative patterns are compared with a new review's graph representation. The review "The discussion covers a lot of different ethical issues." has summative content-edges such as "covers – lot" and "covers – issues", which have a high $\textit{content}_{summative}$ match of 1.19 with the selected patterns. This indicates the presence of summative content in the new review.

# Experiment

Our aim with this work is not to classify a review based on its content type, since a review can contain more than one type of content, but to identify the amount of each type of content (quantified on a scale of 0–1 as shown in Figure 1) a review contains. Content values are provided to help reviewers adjust their reviews with respect to the content types they have received low scores on.

For the purpose of evaluating this patterns-based approach, review segments are tagged based on its predominant content type. For each review the machine selects content type $C$, which produces $\operatorname{argmax}(\textit{content}_C)$ (from Equation 2).

We demonstrate that word-order graphs together with semantic relatedness metrics produce patterns that are better at identifying the content type of a review than classifiers trained on non-trivial semantic features. For our baselines we use the following features: (i) unigrams, (ii) bigrams, (iii) graph edges, (iv) tokens tagged with state, produced during the graph generation process and (v) top 200 topic words[1] that are representative of each content type. Topics are identified using LDA (Latent Dirichlet Allocation) (Blei, Ng, and Jordan 2003). We train the listed features using (A) L1-regularized logistic regression and (B) multi-class support vectors (SVM) as learners.

## Data

We evaluate our approach on peer-review data from Expertiza (Gehringer 2010) and the SWoRD (Patchan, Charney, and Schunn 2009) projects. Expertiza and SWoRD are collaborative web-based learning applications that help students work together on projects and critique each others work using peer reviews.

We evaluate our technique on 1453 academic reviews selected randomly from Expertiza and on 1048 reviews from the SWoRD project, i.e., a total of 2501 reviews. We randomly selected 10% of the reviews from Expertiza and got four annotators to identify their most representative content type. The average inter-rater agreement between the four annotators was 82% and the average Kappa was 0.74 (Fleiss, Cohen, and Everitt 1969). A high Kappa indicates that humans agree on the types of content the reviews contain. The average Kappa between each of the three raters and a fourth rater was 0.75. Because of a high degree of agreement the fourth annotator labeled all reviews, and these labels were used in the pattern learning process.

We obtained annotated SWoRD data from the project's team at the University of Pittsburgh. The dataset has been used by Patchan, Charney, and Schunn (2009) to compare reviews written by students, a writing instructor and a content instructor. The data was coded as summary, praise or criticisms (containing explicit problem or explicit solution) by two human judges. The judges coded the data in two steps: (1) determining the type of feedback (summary, praise, problem/solution) and (2) distinguishing problem and solution reviews. The kappas for each of the coding steps were 0.91 and 0.78 respectively (Patchan, Charney, and Schunn 2009).

In order to combine the two datasets for our evaluation, reviews from SWoRD that are coded as summary or praise are treated as *summative* reviews, and reviews coded as explicit problems are treated as *problem detection* reviews, while those coded as explicit solutions are treated as *advisory* reviews.

The dataset contains a total of 1047 summative, 710 problem-detection and 744 advisory reviews. We look for patterns of the most prominent content type among the reviews. We use a hold out based (splitting) validation, in which the data set is divided into two disjoint sets—training

[1] We select 200 topics since on average there are about 200 tokens in every content type's pattern set.

Table 2: Average recall, precision and $f$-measure for the different approaches (using 5-fold cross validation averages).

| Approach | Accuracy | Precision | Recall | $f$-measure |
|---|---|---|---|---|
| **Patterns** | **67.07%** | **0.68** | **0.66** | **0.67** |
| **SVM, Unigram** | 35.76% | 0.33 | 0.34 | 0.33 |
| **LR, Unigram** | 33.73% | 0.33 | 0.33 | 0.33 |
| **SVM, Bigram** | 31.39% | 0.32 | 0.33 | 0.32 |
| **LR, Bigram** | 35.07% | 0.33 | 0.33 | 0.33 |
| **SVM, edges** | 32.16% | 0.32 | 0.32 | 0.32 |
| **LR, edges** | 35.09% | 0.34 | 0.35 | 0.34 |
| **SVM, tokens+state** | 35.84% | 0.36 | 0.36 | 0.36 |
| **LR, tokens+state** | 36.08% | 0.35 | 0.35 | 0.35 |
| **SVM, topics** | 33.79% | 0.34 | 0.33 | 0.34 |
| **LR, topics** | 34.45% | 0.32 | 0.33 | 0.32 |

**The differences between precision, recall and f-measure values of the patterns-based approach and the classifiers' results are significant (two-tailed test, p-values $< 0.05$, thus the null hypothesis that this difference is a chance occurrence may be rejected).*
**SVM: support vectors, LR: Logistic Regression*

and testing. The training data set is used to identify the semantic patterns. Patterns are used to identify content type of reviews in the test set. We use 1751 reviews for training ($\approx$ 70% of the data) and the remaining 750 for testing. We calculate our final results using a 5-fold cross-validation. During each run patterns are identified from 4-folds of the dataset and tested on the $5^{th}$ fold. The results from the five runs are averaged to get the final results listed in Table 2. Cross-validation ensures that data from both sources go into the training (pattern identification) and testing steps at one point or another.

## Results

Results in Table 2 show that our pattern matching approach produces high precision, recall and $f$-measure values for the task of review content identification. Data distribution across the three content types is as follows: 41.86% summative, 28.38% problem detection and 29.74% advisory. The average distribution across the test sets is 41.31% summative, 28.27% problem detection and 30.43% advisory. Our approach's accuracy of 67.07% is greater than the percentage of the largest content type–summative.

Our approach produces better precision, recall and $f$-measure values than support vectors and logistic regression. Unigrams perform well for both support vectors and logistic regression, and produce higher results than the bigrams-based models. Logistic regression and SVM perform well on edges and tokens tagged with state as features. Logistic regression has its best accuracy of 36.08% and SVM has its best accuracy of 35.84% with tokens tagged with state information as features.

We also tested our approach by identifying patterns for reviews from Expertiza and then testing them on reviews from SWoRD and vice versa. The results are listed in Table 3. The performance of the pattern-based approach is comparable to its performance when reviews from both datasets were mixed to produce the train and test sets. This shows that the generated patterns are generalizable across datasets irrespective of the content of the reviews.

We found that our approach produces high precision and

Table 3: Average recall, precision and $f$-measure obtained when trained on one data source and tested on a different source.

| Approach | Accuracy | Precision | Recall | $f$-measure |
|---|---|---|---|---|
| **Train: Expertiza, Test: SWoRD** | | | | |
| **Patterns** | **62%** | **0.66** | **0.59** | **0.62** |
| **SVM, Unigram** | 31.88% | 0.47 | 0.35 | 0.40 |
| **LR, Unigram** | 41.08% | 0.38 | 0.38 | 0.38 |
| **SVM, Bigram** | 32.16% | 0.35 | 0.29 | 0.32 |
| **LR, Bigram** | 39.78% | 0.34 | 0.34 | 0.34 |
| **SVM, edges** | 31.51% | 0.32 | 0.32 | 0.32 |
| **LR, edges** | 38.94% | 0.34 | 0.34 | 0.34 |
| **SVM, tokens+state** | 30.11% | 0.32 | 0.33 | 0.33 |
| **LR, tokens+state** | 36.80% | 0.35 | 0.35 | 0.35 |
| **SVM, topics** | 43.87% | 0.44 | 0.33 | 0.38 |
| **LR, topics** | 42.84% | 0.32 | 0.38 | 0.34 |
| **% of largest class** | 43.87% | | | |
| **Train: SWoRD, Test: Expertiza** | | | | |
| **Patterns** | **66%** | **0.70** | **0.65** | **0.67** |
| **SVM, Unigram** | 41.64% | 0.42 | 0.41 | 0.42 |
| **LR, Unigram** | 43.43% | 0.40 | 0.40 | 0.40 |
| **SVM, Bigram** | 30.14% | 0.54 | 0.42 | 0.47 |
| **LR, Bigram** | 39.02% | 0.34 | 0.35 | 0.34 |
| **SVM, edges** | 31.52% | 0.31 | 0.32 | 0.32 |
| **LR, edges** | 39.16% | 0.37 | 0.35 | 0.36 |
| **SVM, tokens+state** | 36.41% | 0.36 | 0.35 | 0.35 |
| **LR, tokens+state** | 46.73% | 0.47 | 0.44 | 0.45 |
| **SVM, topics** | 42.81% | 0.40 | 0.40 | 0.40 |
| **LR, topics** | 37.78% | 0.31 | 0.34 | 0.32 |
| **% of largest class** | 41.5% | | | |

**The differences the patterns-based approach's results and the classifiers' results are significant (two-tailed test, p-values $< 0.05$).*

recall values for reviews containing advisory content. At times problem detection reviews tend to get misclassified as summative or advisory reviews. Consider the problem detection review "There are quite a *few grammatical errors* making the webpage *more difficult* to understand than it *should be*." Tokens *more* and *should be* appear often among advisory reviews (see Table 1). As a result the problem detection review is misclassified as an advisory review.

As seen earlier logistic regression and SVM perform better (on accuracy) when trained on more meaningful features such as tagged tokens or topic words than just unigrams or bigrams. Both SVM and logistic regression trained on SWoRD data with topic words and tokens tagged with state as features, respectively, produce accuracies greater than the % of the largest class. This indicates that there might be some specific tokens (topical words) that may help identify content type. These models perform well on the summative class (largest content type), but poorly on the other two content types. As a result, they have better accuracies but lower $f$-measures.

Logistic regression and support vectors perform well when there is a good overlap between the vocabularies of the train and test sets. However, semantics and structural information of reviews play a crucial role in distinguishing between content types. For instance words such as "easy", "great" and "well-organized" are common among the sum-

mative reviews in the training dataset and are weighted highly by the logistic regression models. As a result, problem detection reviews containing negations of these tokens as in "The prose is *not easy* to understand" or "*... well organized* but there are several *grammatical errors* throughout the text ..." tend to get misclassified as summative reviews. Hence, approaches that focus on exact matches of tokens and that do not take context or varying degrees of similarity into consideration may not succeed in capturing patterns that distinguish review content types. Thus despite being trained with informative features such as graph edges, tokens tagged with state and topic-representative words, logistic regression and SVMs do not succeed in accurately determining the content type of reviews.

## Conclusion

In this paper we propose the use of a graph-based text representation to identify structural patterns in reviews containing different types of content. Our work is important because this is a pioneering effort in the application of semantic patterns to the little-explored problem of content identification in academic reviews. Review content identification has potential use in applications that incorporate peer-reviewing as an assessment technique (e.g. MOOCs such as Coursera). Reviews provided by peers may be used to make assessment decisions, and reviews with useful content are likely to be more trustworthy.

The system we propose learns patterns from past reviews, and uses it to identify the content type of new reviews. Our approach is different from conventional approaches that use exact matches of frequent tokens to identify content type. We have shown that: (1) our content identification technique has an $f$-measure of 0.67 on peer reviews from two data sources–Expertiza and SWoRD, and (2) our approach performs better than support vectors and logistic regression learners trained on non-trivial features.

## References


Balfour, S. P. 2013. Assessing writing in moocs: automated essay scoring and calibrated peer review. *Research & Practice in Assessment* 8(1):40–48.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3:993–1022.

Bohnet, B. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 89–97.

Chapman, W. W.; Bridewell, W.; Hanbury, P.; Cooper, G. F.; and Buchanan, B. G. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics* 34(5):301–310.

Cho, K. 2008. Machine classification of peer comments in physics. In *Educational Data Mining*, 192–196.

Coursey, K., and Mihalcea, R. 2009. Topic identification using wikipedia graph centrality. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, 117–120.

Erkan, G., and Radev, D. R. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* 22:457–479.

Fellbaum, C. 1998. Wordnet: An electronic lexical database. *MIT Press* 423.

Fleiss, J. L.; Cohen, J.; and Everitt, B. 1969. Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin* 72(5):323.

Gehringer, E. F. 2010. Expertiza: Managing feedback in collaborative learning. In *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support*, 75–96.

Goldin, I. M., and Ashley, K. D. 2010. Eliciting informative feedback in peer review: Importance of problem-specific scaffolding. In Aleven, V.; Kay, J.; and Mostow, J., eds., *Intelligent Tutoring Systems (1)*, volume 6094 of *Lecture Notes in Computer Science*, 95–104. Springer.

Harabagiu, S.; Hickl, A.; and Lacatusu, F. 2006. Negation, contrast and contradiction in text processing. In *Proceedings of the 21st National Conference on Artificial intelligence - Volume 1*, 755–762.

Liu, B.; Hu, M.; and Cheng, J. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, 342–351.

Manshadi, M.; Gildea, D.; and Allen, J. 2013. Plurality, negation, and quantification: Towards comprehensive quantifier scope disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-13)*.

Mihalcea, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo.

Nelson, M. M., and Schunn, C. D. 2009. The nature of feedback: How different types of peer feedback affect writing performance. In *Instructional Science*, volume 27, 375–401.

Patchan, M.; Charney, D.; and Schunn, C. 2009. A validation study of students end comments: Comparing comments by students, a writing instructor, and a content instructor. *Journal of Writing Research* 1(2):124–152.

Radev, D. R.; Jing, H.; Styś, M.; and Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40(6):919–938.

Ramachandran, L., and Gehringer, E. F. 2012. A word-order based graph representation for relevance identification (poster). *Proceedings of the 21st ACM Conference on Information and Knowledge Management*.

Ramachandran, L., and Gehringer, E. 2013a. Graph-structures matching for review relevance identification. In *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, 53–60.

Ramachandran, L., and Gehringer, E. F. 2013b. A user study on the automated assessment of reviews. In *AIED Workshops*. Citeseer.

Ramachandran, L.; Ravindran, B.; and Gehringer, E. F. 2013. Determining review coverage by extracting topic sentences using a graph-based clustering approach (poster). *The 6th International Conference on Educational Data Mining* 346–347.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In Proceedings of HLT-NAACL*, 252–259.

Tubau, S. 2008. *Negative concord in English and Romance: Syntax-morphology interface conditions on the expression of negation*. Netherlands Graduate School of Linguistics.

Xiong, W.; Litman, D. J.; and Schunn, C. D. 2010. Assessing reviewer's performance based on mining problem localization in peer-review data. In *EDM*, 211–220.