A Dashboard for Analysing Student Feedback Powered by Language Processing

Connor Cassidy, Supervised by Martyn Parker

Department of Statistics, University of Warwick

WARWICK
THE UNIVERSITY OF WARWICK

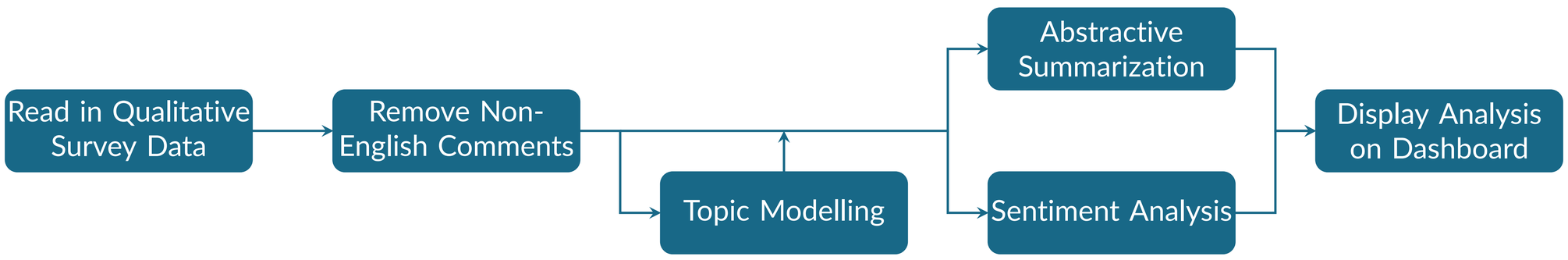
Abstract

Background: Learning Analytics (LA) is a field of research dedicating to improving education quality through leveraging data. Modern advancements in natural language processing (nlp) are facilitating a shift towards the use of large scale qualitative data. To improve the accessibility of qualitative data analysis, we aim to create a dashboard capable of summarizing large numbers of responses to qualitative questions, displaying this information in a meaningful way to educators.

Methods: **Sentiment Analysis** has been used to quantify both the subjectivity and positivity of comments. **Abstractive Summarization** allows the aggregation of overwhelming amounts of free-text responses to a single paragraph. **Topic Modelling** has been used to extract topics from comments. For each topic, Sentiment Analysis and Abstractive summarization has been applied. Additionally, **Extractive Summarization** methods have been applied to help evidence conclusions.

Results: A quarto dashboard was created following the pipeline below. This dashboard allows users to modify hyperparameters, and re-generate output if any is unsatisfactory. This dashboard is made available via a github repository, accessed either by scanning the QR code above or navigating to `https://github.com/Connor-Cassidy/URSS-Student-Feedback-NLP`.

Pipeline for Data Processing



Pre-Processing

- **All Components:** Non-English comments are immediately removed as all models employed are optimized for english text only.
- **Sentiment Analysis:** Both VADER and TextBlob have in build pre-processing, including removal of stop words and un-important punctuation.
- **Topic Modelling:** Punctuation, Stop word, and infrequent word removal.

Sentiment Analysis

VADER[1] (**V**alence **A**ware **D**ictionary for s**E**ntiment **R**easoning) is a rule-based model primarily designed to quantify the positivity of social media text, a context close to that of student reviews.

Key Features:

- **Lexicon** based model assigns sentiment values to commonly used words and phrases.
- **Punctuation and Capitalization** is used to differentiate sentiment intensity between text such as "I love <>" and "I LOVE <>!!".
- **Emoticons and Slang** VADER is built to capture this informal language frequently used by students.
- **Negations and Intensity Modifers** impact the sentiment of following text.

Output: 4 scores for each input, the percentage of the string deemed to be positive, negative and neutral, alongside a compound score ranging from -1 to 1 denoting the overall sentiment of the text.

Model	Acc.	F1	Precision	Recall
VADER	***74%***	***0.83***	0.76	**0.90**
TextBlob	72%	0.80	***0.77***	0.84

Table 1: Performance metrics for VADER against an alternate model, TextBlob. Models were evaluated on 14,607 (67% positive) module evaluation style comments.

To quantify the subjectivity of comments, we use the python package **TextBlob**. This model is conceptually similar to VADER, though it is not optimized for social media text.

Topic Modelling

Latent Dirichlet Allocation[2] (LDA) is an unsupervised, generative model. Under LDA, we assume each comment c is generated via the following process:

1. Generate the number of words, $N \sim \text{Poisson}(\xi)$.
2. Generate the topic distribution, $\theta \sim \text{Dirichlet}(\alpha)$.
3. For $i = 1$ to N :
   (a) Choose a topic z_i using θ.
   (b) Generate a word w_i from $\rho(w_i|z_i, \beta)$.

Here, dim(α) = dim(θ) = Number of Topics is a user defined hyper-parameter. The goal of LDA is to re-construct the distribution of $\rho(w|z_i, \beta)$ used to generate the words for each topic. To re-construct this random variable, Gibbs Sampling is employed through use of the python package **tomotopy**.
In addition to providing a framework for LDA, tomotopy provides methods to construct labels for each topic by maximising pointwise mutual information (PMI). This process involves extracting sections of comments labeled as the current topic and maximising the PMI of n-gram subsets of these comments. This method is a form of **Extractive Summarization**.

Extractive Summarization

TextRank[3] is a graph based ranking algorithm which we use to extract the most important comments. Here, a comment is deemed 'important' if the meaning of the comment appears frequently in other comments. The algorithm is as follows:

1. Compute a similarity matrix across all comments.
2. Construct a graph based on this matrix, whereby each comment is a node and each edge is a similarity score.
3. Rank node importance using **PageRank**.
4. Return the top n most important comments.

To construct the similarity matrix, we use the cosine similarity between sentence embeddings. These sentence embeddings are 384-dimensional vectors describing the meaning of each comment, computed using **all-MiniLM-L6-v2**.

Abstractive Summarization

Abstractive Summarization involves condensing text into a summary, expressing its core ideas in a new way.
Transformers are a class of neural networks used in nlp for tasks like abstractive summarization. Their inclusion of 'self-attention' enables entire text sequences to be processed at once.
BART[4] is a transformer model that processes input text bidirectionally, capturing context of each input word from both sides. Its decoder outputs text autoregressively, where each output word depends also on the previous output words.
The **T5 Text-to-Text Transformer**[5] has a more versatile framework, whereby each problem is framed as text generation. T5 leverages extensive training data, with its largest size containing 11 billion parameters.

Extensions

- **Topic Modelling:** This could be improved through automatic selection of the number of topics (such as the Heirarchical Dirichlet Process).
- **Sentiment Analysis:** To calculate the sentiment around topic i, the overall sentiment of each review is weighted by the proportion of topic i in that topic, to produce an overall sentiment score. To improve this, sentiment should be measured about a given topic for each review, then each review sentiment weighted by the confidence in that topic appearing.
- **Abstractive Summarization:** This could be improved by upgrading the use of text-to-text transformers to large language models.

References

(1) C. Hutto and E. Gilbert, *Proc. Int. AAAI Conf. on Web Soc. Media*, 2014, **8**, 216–225.

(2) D. Blei, A. Ng et al., 2001, vol. 3, pp. 601–608.

(3) R. Mihalcea and P. Tarau, Proc. EMNLP Conf. Ed. D. Lin and D. Wu, 2004, pp. 404–411.

(4) M. Lewis, Y. Liu et al., ed. D. Jurafsky, J. Chai et al., Ass. for Computational Linguistics, 2020, pp. 7871–7880.

(5) C. Raffel, N. Shazeer et al., *J. Mach. Learn. Res.*, 2020, **21**.