Journal of Computational Social Science (2025) 8:43
https://doi.org/10.1007/s42001-025-00366-y

RESEARCH ARTICLE

Stance classification: a comparative study and use case on Australian parliamentary debates

Stephanie Ng[1] · James Zhang[1] · Samson Yu[2] · Asim Bhatti[1] · Kathryn Backholer[3] · C. P. Lim[1]

Received: 30 July 2024 / Accepted: 2 February 2025

Abstract

Hansard, or the official verbatim transcripts of parliamentary debates, contains rich information for analysing discourse and political activities on a wide range of policy issues. A fundamental task in political text analysis is to predict whether a speaker takes on a positive or negative view about a debate topic. Unlike social media data, which has received extensive attention for political text mining, stance analysis on Hansard data remains understudied. The main distinctions between the two include longer text and context dependency related to a motion in the Hansard data. As a result, it is difficult to devise a text mining model for parliamentary debates based on existing studies of other applications. This raises the question of the generalisability of prominent methods for cross-domain classification under low-resourced data situations. To address this issue, we construct and compare various state-of-the-art natural language processing techniques and machine learning models for stance classification, using two benchmark datasets from the UK Hansard. To improve the model accuracy, a hybrid approach is designed, which leverages both text and numerical features in the classification process. The devised method achieves 15–20% improvement in accuracy compared to the baseline methods. Transfer learning of pre-trained language models is further investigated for political text representation and domain adaptation in a new stance classification task: Australian Hansard with debates focusing on the public health issue of obesity and related junk food marketing policies. Then, a feature augmentation technique is employed to optimise the learning model from the source domain for prediction on unseen test data in the target domain. This approach results in approximately 10% improvement in accuracy compared to those from the baseline methods. Finally, an error analysis is conducted to gain further insights into the devised model, which reveals the characteristics of commonly misclassified samples and suggestions for future work.

Keywords Stance classification · Transfer learning · Domain adaptation · Natural language processing

Extended author information available on the last page of the article

Published online: 03 March 2025

1 Introduction

There has been increasing interest in 'text-as-data' across domains such as media communication, political science, and the social sciences in general, enabling data-driven analysis of text documents through state-of-the-art natural language processing techniques and machine learning models [1–3]. An important data source for political texts analysis is the Hansard (official verbatim transcripts of parliamentary debates), which contains a plethora of information for analysing discourse and political activities on a wide range of policy issues. Analysis of how policy issues are portrayed by politicians during parliamentary debates can be beneficial in many practical applications. On the one hand, it is useful for researchers and advocates to derive insights to better understand controversial debate topics. On the other hand, it is important for policy makers to monitor communications and political activities on priority issues to formulating strategies that effectively convey or frame policy actions.

A fundamental task in political text analysis is to identify the political stance of a speaker, which is to predict whether they take on a positive or negative position towards a targeted issue [4]. Such predictions enable the identification of political homophily and ideology among speakers, as well as early detection of potential policy development. Previous studies on stance classification largely focus on congressional floor debates [5] and online debates [6]. Since the publication of relevant tasks in SemEval2016 [7], most studies now focus on Twitter data analysis, as well as on the news domain, with attention to Fake News [8] and Rumor detection [9] (see [10] for a survey on stance detection). With the digitization of parliamentary transcripts in countries such as Canada [11, 12], the United Kingdom [13–16], as well as open access to parliamentary corpora of European countries through the CLARIN infrastructure [17], recent work in stance classification has shifted its attention back to social media posts and online debate forums pertaining to parliamentary debates. Various computational approaches and statistical text analysis methods have been widely adopted in many countries, in contrast to Australia where a limited number of literature can be found. The most relevant study that we have found is the work in [18]. which adopted CoreNLP sentiment analysis and Latent Dirichlet Allocation topic models to analyse transcripts of Australian Prime Ministers. The unsupervised approach to analyse political framing was adopted to compensate the lack of labelled datasets, but there is still a major gap in the interpretability of a topic model and their inference of the political stance of a speaker. The need for this model was attributed to the complex political language, the lack of labelled data for model training, and the fact that transcripts are available only in PDF or HTML file formats, which require laborious data pre-processing tasks to obtain a machine-readable input dataset.

As a newly emerged research area in the computational social science domain, stance classification is often used interchangeably with sentiment classification for political text analysis, as well as other terms used in the literature such as support/opposition detection, perspective detection, or agreement detection. The diversity and inconsistency in terminology are deemed problematic

and challenging for political text mining research [19], as the tasks and scope of analysis become unclear to the readers. While both are closely related, it has been shown that, in the case of political texts, stance and sentiment do not necessarily align, since a stance is multifaceted and involves more complex use of language in argumentation to convey opinions, rather than just positive or negative notions [20, 21]. Yet, it has been shown that, in the case of Twitter data and online debates, sentiment information can be beneficial for stance classification with the assistance of an "opinion-towards" label [6, 22]. Here we consider sentiment analysis as a sub-problem of stance classification at a macro-level due to the lack of such label information.

The primary objective of this study is to develop a high-performing machine learning model for stance analysis that can effectively generalise to unseen data. We use political text classification as a practical application domain to test our model, situating our research within the broader contexts of text mining and natural language processing. Specifically, we focus on the fundamental stance classification task to detect the position of speakers, whether they support or are against a policy in a parliamentary debate, using two UK Hansard datasets available publicly. To identify the best-performing model for stance classification, a comparative study is presented to examine the effectiveness of language models on classifying unseen samples and unseen topics in political texts under different settings. Traditional machine learning and deep learning-based pre-trained transformers language models are evaluated to identify one that can best represent political texts. We identify the best-performing stance classification model through several criteria, including different training data sizes, sequence lengths, text representations, as well as classification algorithms (Support Vector Machine and Gradient Boosting Classifiers) and their hyperparameters. We also fill the gap on the under-developed computational approaches pertaining to Australian Hansard text analytics. To cope with the lack of labelled data in our target domain, we further investigate domain adaptation techniques to bridge the distribution gaps of target data from the source domain. The proposed approach consists of three fundamental steps that integrate feature design and evaluation methodology:

1. A hybrid approach that capitalises on both textual and numerical features is created, employing traditional machine learning-based ensemble algorithms alongside large language models. We achieve superior classification performance by applying this approach to two published data sources.
2. A rigorous group-splitting strategy for model evaluation and selection is adopted, in contrast to the random splitting method commonly found in the literature. This approach helps us avoid data leakage and overfitting while identifying the best performing models.
3. To optimise our learning models for accurate predictions on unseen target domain data, a feature augmentation technique is employed from domain adaptation in transfer learning. This is based on a dataset from the Australian parliamentary debates.

2 Literature review

2.1 Stance classification models

For undertaking supervised stance classification on parliamentary debates, the initial work in [5] modelled the relationships among speech segments using the ConVote dataset. Subject to the same-speaker constraint, it was found that concatenation of utterances by the same-speaker improved the model performance. For the different-speaker agreements, agreement links with text surrounding target of references were computed. However, the stance classification task is evaluated independently of a motion, i.e., the speaker's opinion was unknown.

For motion-dependent stance classification, many existing methods utilise either the TF-IDF (Term Frequency-Inverse Document Frequency) word features with traditional machine learning models or word embeddings with the neural network models [14, 23, 24]. Most work has focused on the Support Vector Machine (SVM) classifier without any tuning. Here we present our comparative study using nested cross validation to evaluate the model performance with optimal parameters based on the gradient boost classifier. Unlike the SVM classifier, which relies on kernel selection to separate datapoints into classes, the gradient boost is a tree-based ensemble model of boosting type. A stronger model can be created by combining many weak learners through iterative fitting of a new model based on residual errors of the prior model. It was found that in many text classification tasks, including hate speech classification [25], spam classification [26, 27], sentiment classification [28–30], and other applications [31–33], the gradient boost classifier consistently outperformed the SVM classifier.

Another research direction in stance classification is based on transfer learning by fine-tuning a transformer model, which has achieved state-of-the-art performance on many NLP (natural language processing) tasks. Although the study in [23] utilises the state-of-the-art BERT (Bidirectional Encoder Representations from Transformers) language model, the authors did not find a significant performance gain from the model in their experimentation. In this study, we examine this issue by carefully designing our training strategies during fine-tuning, as well as exploring two BERT variants: DistilBERT and RoBERTa. On the other hand, the work in [24] and [34] proposed a graph-based method to better capture the interdependent relationships among speakers, motions, and speeches in parliamentary debates. However, their feature sets included the use of speaker party affiliation, motion party affiliation, and debate identification (ID), leading to difficulties in model adaptation for unseen debates and unseen domains classification tasks. We examine the possibility of other non-text features. The authors of [35] and [36] found a performance gain when including document statistics, the number of repeated punctuation symbols, syntactic dependency features, and its preceding post features on top of n-grams features for classifying stance in online debate. Besides that, [37] leveraged user information and user posting behavioural features for sentiment classification of policy microblog, [38] incorporated stance polarity and intensity labels, while [39] focused on the user's interactions to improve stance prediction.

2.2 Cross-domain transferability

There is a growing interest in performing cross-domain stance classification tasks. In the case of in-domain classification, the test set is randomly selected from the training data of the same distribution. Researchers in [40] argued that such a setting is not always valid in real-world applications. In fact, limited work has been done on stance classification of online debate forums, such as [41] and [42], where model training with group splits on datasets is performed based on the debate topics. On the other hand, previous studies on parliamentary debates [14, 23, 24, 34] adopted a random split approach on datasets, which allows datapoints belonging to the same group, i.e., motion from the same debate, to appear in both the training and test sets. In comparative studies of stance classification models, the existing literature mainly focuses on social media and online debate data [43, 44]. The studies in [45] and [46] investigates the potential of multi-dataset learning to improve the robustness of stance classification models with datasets from various domains. Other studies have explored stance classification with opinion-towards labels, i.e., in multi-target and cross-target stance classification [47–49]. In particular, the authors in [50] proposed a BERT model with adversarial learning and knowledge distillation for domain adaptation in cross-target stance.

To improve model performance for cross-domain classification, researchers in [51] proposed domain adaptation for aspect-based sentiment classification on review datasets using self-supervised fine-tuning of a BERT model. A 2-3% drop in accuracy was observed for cross-domain training compared to in-domain training. Additionally, the work in [52] adopts the feature augmentation technique proposed by [53] for stance classification of COVID-19 tweets. In this case, the incorporation of such a technique results in the best performing model, compared to direct inference using only a limited number of in-domain target data, cross-domain source data, or all available annotated data. Moreover, TrAdaBoost is proposed in [54], a boosting method for instance-level domain adaptation. Experiments in [55] and [56] have shown promising results in their respective classification tasks for applications in sentiment classification and fraud detection. However, the applicability of these approaches remains unclear in stance classification. One significant challenge in this regard is the need to acquire a domain-independent representation while avoiding the risk of negative transfer [57]. To address this gap and gain insights into achieving optimal cross-domain transferability in stance classification, particularly within a political context, this study investigates domain adaptation techniques spanning instance, feature, and parameter-level adaptations.

While great progress has been made in stance classification, the use of Hansard datasets for stance classification is rare, unlike social media and online debate datasets. A survey in [4] reveals that only 4 out of 17 articles are related to political stance detection from 2006-2016. A large part of previous work on political text mining has focused on sentiment analysis or stance classification using Twitter datasets [40, 58, 59]. Besides that, only 3 out of 12 datasets were found to be related to parliamentary debates [4], including one debate transcript from US: ConVote [5], and two from the UK: HanDeSet [14] and ParlVote [23]. One potential reason is the challenge associated with Hansard data analytics, which includes longer texts,

context dependency on a motion, relevance to preceding conversations, and even continuation from earlier debates [60]. As a result, we argue that stance classification pertaining to parliamentary debates is highly complex, and this task is undertaken in our study.

3 Methodology

3.1 Problem statements

3.1.1 Stance classification

A parliamentary debate often begins with a Member of Parliament (MP) moving a motion, followed by other members speaking in turn, presenting their viewpoints, questioning, or even proposing an amendment to counter the motion. Each speaking turn is called an utterance. At some point during the debate, typically towards the end, a division may be called for members to register their votes. These votes provide a good indication of a speaker's stance, which can be used as labels in stance classification. Moreover, the process of policymaking and the context of debates can evolve over time. Having an effective tool that can provide a timely insights into the current political stance in such dynamic and complex environments is useful for informed decision making. The outcomes may also be valuable for advocacy related to a particular issue and for future political text analysis. These motivations form the basis of our research. Specifically, we define the stance classification task in the context of parliamentary debates as follows:

Given a transcript consisting of a motion, member speeches (i.e., the concatenation of utterances by speakers), and metadata related to the speakers and debates, a stance classification task aims to classify each speech in the transcript with a label $y \in \{1, 0\}$, where 1 represents a positive stance and 0 represents the opposite stance of the particular speaker towards the debate motion.

3.1.2 Domain adaptation

The main challenge in our research is the lack of a labelled dataset for model training, particularly for the Hansard data. Although these datasets are openly accessible via government websites for each state, large scale data cleaning for Australian Hansard remains a challenging task, as the transcripts are only available in PDF or HTML file format. In the absence of metadata or pointers to attribute values, it is difficult to convert unstructured raw text into structured tabular data for training machine learning models. As an example, pre-processing texts into machine readable inputs requires several subtasks, such as identifying segmentations in a transcript at both the debate-level and speaker utterance-level, as well as extracting information such as debate title, date, speaker names and political party affiliations. Moreover, different states in Australia have different layouts and text styles in their Hansard compilation. Designing a generic process that can capture such differences while maintaining a high-quality, error-free corpus is

cumbersome. To facilitate our current research, we perform a goal-directed search on a topic of interest and create an Australian Hansard suitable for analytics with machine learning models. An alternative approach to creating a new annotated corpus is transfer learning, which leverages knowledge from existing tasks to optimise a related new task. The most relevant data source for stance classification we have found are the UK Hansard [14, 23]. We exclude the ConVote [5] US Hansard from our experimentation because the motion text is not available. Since Australia is part of the British Commonwealth nation and shares a similar system of government, we investigate whether it is possible to leverage the existing UK Hansard datasets for our target dataset through domain adaptation and transfer learning techniques. We formulate this domain adaptation problem as follows:

Consider a large labelled source domain dataset $D_S = \left(x_i^s, y_i^s\right)_{i=1}^{N_S}$ and a small labelled target domain dataset $D_T = \left(x_i^t, y_i^t\right)_{i=1}^{N_T}$, where S denotes the source domain, T denotes the target domain, and N_S and N_T represent the respective number of samples in each domain. The data samples in each domain are drawn from domain-specific distributions: $x_i^s \sim P_s(x)$ for the source and $x_i^t \sim P_t(x)$ for the target domain. The objective is to derive an optimal classification model for unseen and unlabelled data $D_U = \left(x_i^u, y_i^u\right)_{i=1}^{N_U}$, which is sampled from the target data distribution $P_t(x)$.

3.2 Datasets

3.2.1 Benchmark/source datasets

To derive an optimal stance classifier, we train and evaluate several models using two benchmark datasets: HanDeSet and ParlVote [14, 23]. Following the procedure outlined in [23, 34], we randomly split the ParlVote dataset into 5 subsets to investigate the impact of training data size and sequence length (512) on classification performance, as follows:

- Small-Any: consists of 1,251 random samples.
- Small-512: consists of 1,251 samples with 512 or fewer tokens.
- Medium-Any: consists of 18,253 random samples.
- Medium-512: consists of 18,253 samples with 512 or fewer tokens.
- Large-All: consists of all 33,461 samples.

Since the ParlVote dataset is slightly imbalanced, we apply stratification to maintain the original class proportion in the subsets, ensuring that the resampled data accurately represents the true population. The sample size for each class label in each subset can be obtained as follows: $n_0 = n \times \frac{N_0}{N}$; $n_1 = n - n_0 = n \times \frac{N_1}{N}$ where $n = n_0 + n_1$ is the total number of samples in the subset and $N = N_0 + N_1$ is the total number of samples in the full dataset, or in the filtered dataset with 512 or fewer tokens.

3.2.2 Application/target dataset

To evaluate the cross-dataset performance of stance classifiers, we focus on debates related to obesity and food marketing policies in Australia. While various legislative bans on unhealthy food marketing have been proposed to address the public health challenge of obesity in Australia, no such policy has been enacted into law to date. To better understand the political representations of food marketing policies and the positions of parliamentarians, we collect debate transcripts from Australian state parliamentary websites.

Figure 1 illustrates our target data creation workflow. A web scraper script was implemented to automatically download search results retrieved in June 2022 for 6 keywords, covering transcripts from 1/1/2000 to 1/1/2022. Due to the difficulty of obtaining transcripts from dynamic web pages without a direct HREF link, the Northern Territory (NT) is excluded from our analysis, leaving 6 Australian states and one territory. The raw transcripts are stored separately for further processing to derive machine-readable data input.

Data extraction involves the following process. Firstly, all PDF files are converted into machine-readable string text documents using PyMUPDF. Next, text segmentation for multiple debates within a transcript is performed by recognising the associated bold debate title. For each debate, the date, debate title, speaker names and their utterances are extracted using regular expressions and rule-based methods. Dates and names with titles are processed into a consistent format. To create a similar feature space as the source dataset, political party information is obtained by mapping results from Wikipedia searches for each speaker. The first utterance of each speaker in a debate, starting with 'I move', is assumed to be the motion of the debate. Then, stance labels are assigned for transcripts that include a voting outcome at the end of the debate. These labelled records are used as data samples for training semi-supervised domain adaptation models in stance classification. Finally, a condition check is performed to remove irrelevant or incomplete records, followed by the merging of records from all states and territories, and grouping utterances of each speaker into speeches to obtain a final data frame for analysis. The collected data can be accessed at [61]. Table 1 summarises the key statistics of the derived datasets.

3.3 Text representation

In this paper, we focus on the following text representations:

- *TF-IDF* calculates the weight of each word in a document as follows [62]:

$$W_{t,d} = tf_{t,d} \times \log\left(\frac{N}{df_t}\right) \tag{1}$$

where $tf_{t,d}$ denotes the frequency of the term in a document, N represents the total number of documents in the corpus, and df_t is the number of documents that contain the term. Research has demonstrated the effectiveness of static

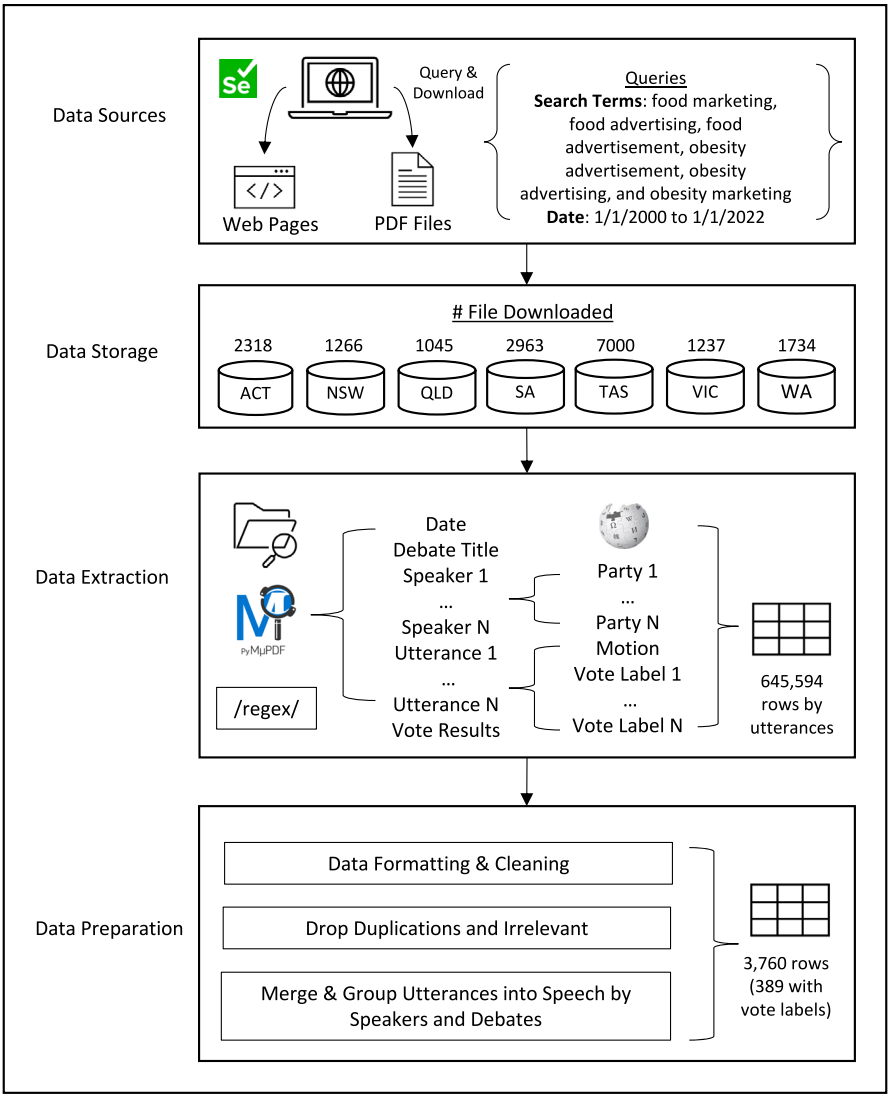

Fig. 1 Data creation workflow

Table 1 Key statistics of dataset

	HanDeSet	ParlVote	AusHansard
Speeches	1,251	33,461	389
Debates	129	1,995	34
Speakers	607	1,346	296
Parties	13	16	22
'Aye' label	713	17,721	210
'No' label	538	15,740	179

embedding methods like TF-IDF, particularly in applications like sentiment classification and long-text analysis, when compared to more advanced embeddings such as Word2Vec [63–65].

- *BERT* is a pre-trained masked language model that can be fine-tuned for downstream tasks such as text classification [66]. Instead of building a new model from scratch, fine-tuning a Transformer that has been trained on a large corpus proves to have a significant positive impact on model performance. The vector representation of each token is derived through the attention mechanism:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

  where Q, K, V represent the Query, Key, and Value, respectively, which are obtained through separate linear transformations from the token. d_k denotes the dimension of the key vector, used for normalisation. Since the output of the attention head is a weighted sum of the value vectors across all representations in the sequence, it enables the extraction of a contextualised representation of the token.
- *DistilBERT* is a lightweight version of the BERT model with 40% fewer parameters. As a trade-off, DistillBERT runs 60% faster, though at the cost of some model performance [67].
- *RoBERTa* (Robustly Optimised BERT Pretraining Approach) is an optimised version of BERT, with improvements to key hyperparameters such as dynamic masking, batch size, and data size [68].

Despite the introduction of more recent large language models like GPT-4 and Llama, recent research has revealed their high sensitivity to prompts, which may hinder their ability to generate the intended responses for classification tasks [69–71]. Considering the computing resource requirements and accessibility to their established performance, this study opted to utilise BERT-based transformers to ensure model comparability and replicability.

3.4 Pre-processing

We preprocess both motion and speech texts by applying lemmatisation along with lowercasing, removal of punctuation, digits, stop words, and words with fewer than three letters. This ensures that all words are converted into their base form before generating the word-level TF-IDF vectors, helping to address sparsity in the representation. For encoding the text with a pretrained language model, we retain the original text to preserve contextual information. The raw text sample are passed directly into a WordPiece [72] tokeniser to generate subword tokens. Besides that, special tokens are added to the text input to help the model recognise the input sequence type. For BERT and DistilBERT, }[*CLS*]′ is added at the beginning of each sequence input for text classification, while }[*SEP*]′ is used as a separator for pairs of sequences, i.e., speech and motion text samples. For RoBERTa, we use ‘s’ and ‘\s’.

3.5 Feature extraction

The following features are obtained to supplement stance classification:

- *Sentiment polarity*: Motion and speech texts with similar sentiment polarity scores may indicate alignment in argumentation, suggesting a positive stance or otherwise. We apply VADER [73], a lexicon-based sentiment analysis tool to output a compound score between -1 and 1 for both motion and speech text samples.
- *Number of words*: The number of words is to account for the length of a speech before pre-processing. This information is not captured in the TF-IDF representation due to the removal of stop words and rare words when building the vocabulary, and sentence truncation in pretrained language models due to limitations on maximum input length.
- *Number of question marks*: The number of questions in a speech can indicate opposition or a sceptical view on a debate topic. Research has shown that different types of questions, including assertive, rhetorical, or challenge questions, play a significant role in shaping discourse, often signaling challenges or opposition [74, 75].
- *Number of negative words*: We obtain an approximation of the number of negative words in a speech by identifying words that start with the following prefixes: 'dis', 'im', 'in', 'ir', 'il', 'non', 'un', and 'not'.
- *Similarity score*: The similarity score is computed as the length of intersection set divided by the length of the total unique words between the motion text and speech text.
- *Same party*: Party affiliation is an important feature, as speakers from the same party tend to show homophily and agree with each other [76]. While previous work has included other metadata features such as debate ID, speaker party affiliation, and motion party affiliation [14, 24, 34] in modelling, these features are difficult to adapt to our case due to unique non-recurring nature of an ID and diverse classes of political parties in the UK and Australia. Instead of using the party information directly as a categorical variable, we use "1" to represent the condition where both the speech speaker and the motion speaker are from the same party and "0" otherwise.

3.6 Model pipeline

Figure 2 illustrates a pipeline for modelling the hybrid features to capture lexical, contextual, numerical, and metadata categorical features, followed by the modelling of stance classifiers and the application of domain adaptation techniques for our target datasets.

The modelling of our stance classifier is implemented using Scikit-learn (sklearn) [77], a Python module for machine learning. In preparation of the input data for training traditional machine learning models, we concatenate all non-text

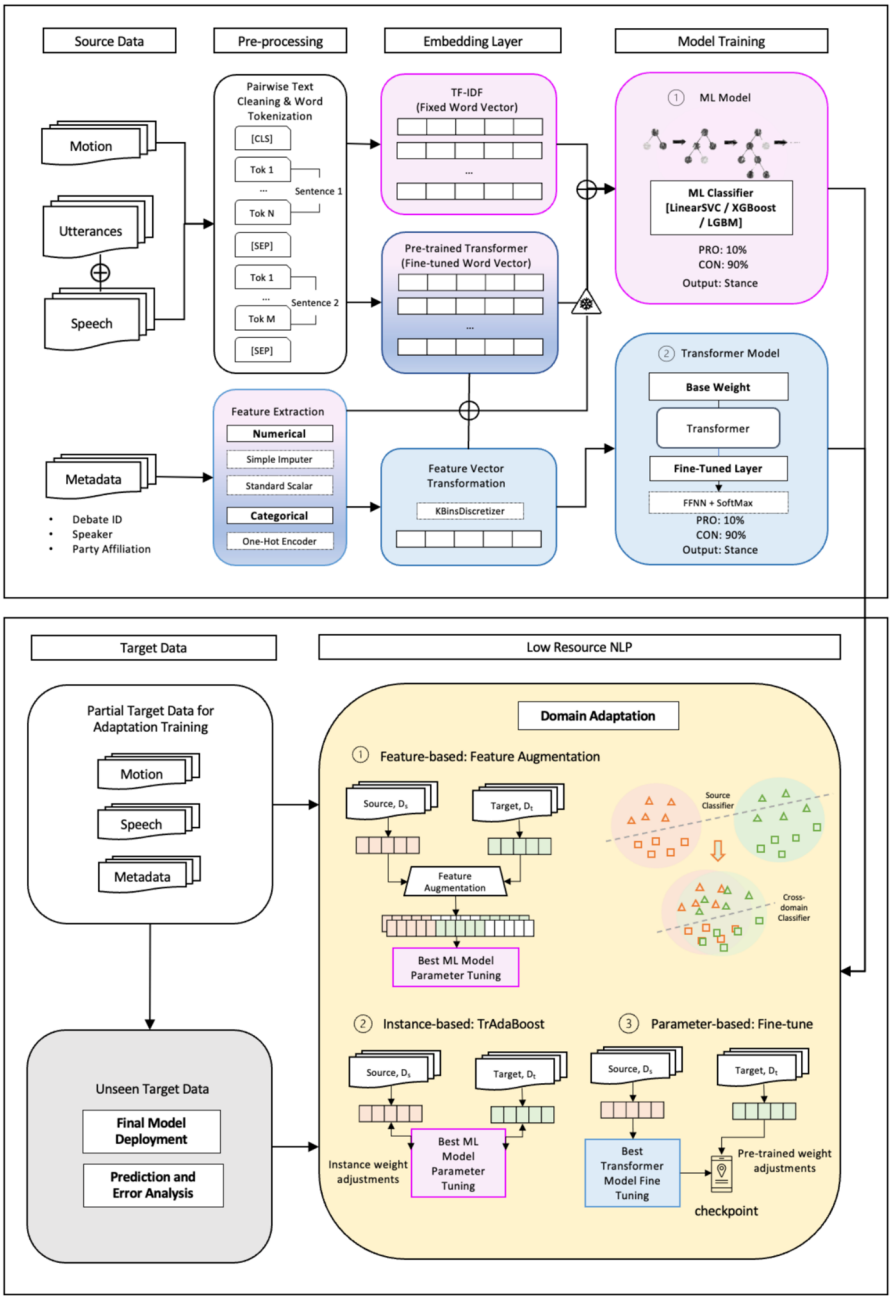

Fig. 2 Model pipeline

features with the TF-IDF representation of the motion and speech. We also obtain the contextual embedding from the last hidden state in the BERT transformer model and apply FeatureUnion to combine the transformer objects, which outputs a new transformer with the concatenation of the BERT embedding and local feature vectors.

For fine-tuning task, we obtain the pre-trained models from Hugging Face's Transformers library [78] and train our model in Tensorflow with the Keras API [79]. We combine non-text features with text inputs using their corresponding special tokens after converting the continuous numerical values into bins of categories with arbitrary label names using the sklearn KBinsDiscretizer method. The transformed result is encoded as an ordinal integer value. We apply a uniform strategy so that the width of all bins is identical.

3.6.1 Stance classifier

Next, we evaluate the following classifiers for our stance classification model:

- *Support vector machine (SVM)*: The SVM categorises two classes by searching for the maximum margin between hyperplanes. The classification task of the SVM with a soft margin is formulated as [80]:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\max(0, 1 - y_i \cdot \hat{y}_i)^2) \tag{3}$$

  where the margin is maximised by minimising weight w, and C is a regularisation parameter with a squared hinge loss function on the right.
- *Gradient boosting classifiers*: Gradient boosting is an ensemble method that builds sequential models based on errors of previous models to minimise the overall prediction error [81]. It is known for its efficiency and low computation requirement. We compare XGBoost [82] and LightGBM (LGBM) [83] in our stance classification task. The objective function is:

$$\min \ \mathcal{L}(\emptyset) = \sum_{i=1}^{n} l(\hat{y}_i, \ y_i) + \sum_{k=1}^{n} \Omega(f_k) \tag{4}$$

  where l indicates the loss function, and $\Omega(f)$ is a regularisation term.
- *BERT-based classifiers*: Building on the BERT-based word embeddings discussed in the text representation section, we adapt the final dense layer of the transformer for our classification task. Transformers, or pre-trained learning models, are neural network-based models composed of multiple encoder-decoder layers and self-attention heads. Fine-tuning is performed using checkpoints from the Hugging Face's Transformers library [78], where the model weights are updated based on the task-specific objective. The final output layer of the network produces logits, which are raw predictions before normalisation. These logits are then passed through a SoftMax function to convert them into probabilities.

The model is then trained using the sparse categorical cross entropy loss function, which is formulated as:

$$\frac{1}{k}\sum_{i=1}^{k}(y_i \cdot log\hat{y_i} + \left(1 - y_i\right) \cdot \log\left(1 - \hat{y}_i\right)) \tag{5}$$

where k=2 for binary label.

The following baseline models are compared in our experimentation:

- *Dummy classifier*: We use the default 'Prior' strategy to make predictions based on the most frequent class label in training dataset.
- *Unigram TF-IDF + LinearSVC*: The same setups as used in [14, 23] is applied to reproduce the best-performing model. The original text without lowercasing is used to produce the TF-IDF representation with unigram features.
- *Deepwalk & GPolS* from [34]: For comparison with published results, we replicate the model performance scores of two graph-based models.
- *BERT*: The BERT base cased model is employed for fine-tuning based on text-only features.
- *Trigram TF-IDF + LinearSVC*: To verify whether text pre-processing and a higher n-gram can improve stance classification, we evaluate on lemmatised text with lowercasing, removal of punctuations, digits, stop words, and words with fewer than 3 letters.

For our models incorporating non-text features, we compare the following eight combinations, which include four machine learning models: Trigram TF-IDF + LinearSVC, Trigram TF-IDF + XGBoost, Trigram TF-IDF + LGBM, Trigram TF-IDF + BERT Features + LGBM; and four pre-trained models from Huggingface: BERT base cased, BERT base uncased, DistilBERT base uncased, RoBERTa base cased.

3.6.2 Domain adaptation

Figure 3 depicts the input distribution of the source and target datasets after dimensionality reduction using t-SNE (t-distributed Stochastic Neighbor Embedding) [84]. From the visualization, we observe noticeable differences between the source and target datasets. Specifically, data samples from the source domain tend to cluster more closely around the central region, while samples from the target domain show a more spread-out distribution.

Table 2 shows the pairwise Maximum Mean Discrepancy (MMD) distances between the datasets, calculated using RBF kernel. The MMD distance between the two UK datasets is smaller compared to the distances between the UK datasets and the AusHansard datasets.

To address the distribution gap for domain adaptation on target dataset, we use two source datasets: HanDeSet and ParlVote to evaluate the model performance without learning from the target dataset, serving as our baselines. Since the source datasets contain significantly more training samples than the target dataset, we use

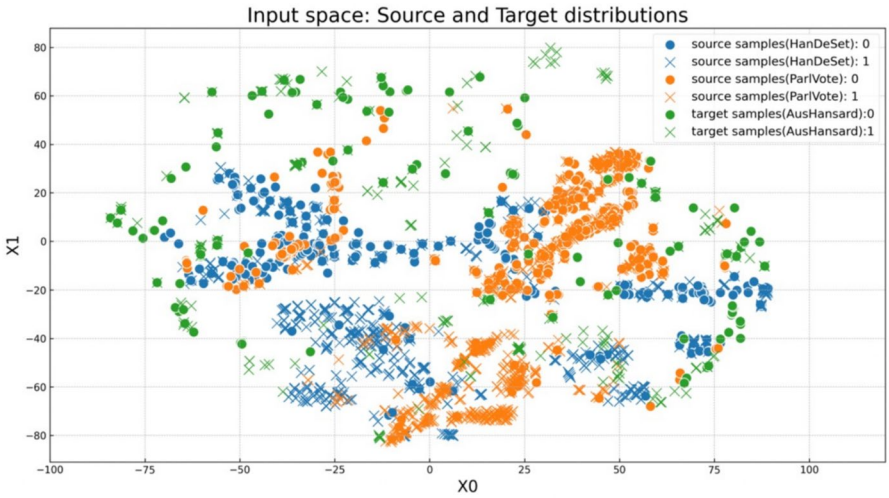

Fig. 3 Source and target distribution

Table 2 MMD between datasets

Dataset 1	Dataset 2	MMD
HanDeSet	ParlVote	0.1248
HanDeSet	AusHansard	0.1587
ParlVote	AusHansard	0.1930

the Small-Any subset from ParlVote to prevent our target domain information from being "washed out" during training.

Next, we explore adaptation techniques across three levels: instance, feature, and parameter.

- *Transfer AdaBoost*: We apply the instance-based domain adaptation technique based on the "reverse boosting" principle, where different weights are assigned to training instances [54]. During each boosting iteration, the weights of source instances that predict poorly are reduced, minimising their impact on the model.
- *Feature augmentation*: Feature augmentation techniques in [53] are utilised to obtain general features covering both source-specific features and target-specific features. The augmented feature vectors include: commonality between the source and target features, source-specific features, and target-specific features, which are then passed to a supervised classifier for stance classification.
- *Parameter fine tune*: In this approach, we replace the output layer of the pre-trained model with a new layer designed for stance classification. Initially, the model is fine-tuned on the source datasets. After that, we continue fine-tuning the model on our target dataset, starting from the checkpoint of the pre-trained model.

4 Experimentation

4.1 Setups

All experiments are conducted on an Apple M1 Max, featuring 10-core CPU, 32-core GPU, and 64 GB of unified memory.

To obtain reliable generalisation error estimates for our models with their optimal hyperparameters, we employ a nested cross validation (CV) procedure. This approach helps mitigate performance bias and overfitting issues during the hyperparameter selection process, ensuring more robust and unbiased evaluation of our models.

Under the 'flat CV' setting, the optimal model is selected based on the best score evaluated from a single validation set and applied to the test set, which can result in an over-optimistic score by sheer chance especially for small datasets [85]. The nested CV procedure, on the other hand, considers the variance from the validation sets by including an inner loop CV to search for the optimal hyperparameters, as well as an outer loop CV to evaluate the model performance based on the test set using the model trained with the optimal hyperparameters, yielding an unbiased estimates of model performance. Moreover, while [86] found that the nested CV could be overzealous in most applications, they were less confident for gradient boost classifiers such as XGBoost, which has more hyperparameters to tune than the traditional classifiers. The training instances are assumed to be independent and identically distributed.

For the purpose of benchmarking the models with the existing work, we use the k-fold CV strategy for the outer loop and the shuffle split CV strategy for the inner loop, producing 10 samples of training, validation, and test sets with the ratio of 80:10:10. Although the standard nested CV procedure utilises double k-fold CV, we use shuffle split CV for the inner loop so that we can configure a specific size for the validation set, i.e., the ratio of 80:10:10, for a fair comparison with previous work. Stratification is also applied during resampling to maintain the original class proportion in the subsets, so that the resampled datasets represent the true population.

The nested CV procedure incurs a high computational cost. For m models with K_1 test sets for outer CV and K_2 validation sets for inner CV on p parameter sets, we fit the model $m \times K_1 \times K_2 \times p$ times. For traditional machine learning classifiers, we parallelise the inner loop with 10 shuffle splits. For transformer models, however, we use predefined hyperparameters and default settings due to limited computing resources. The hyperparameters are selected based on preliminary trials using KerasTuner with Bayesian Optimisation [87]. We obtain 1 shuffle split for the validation set instead of monitoring the validation accuracy and calling for early stopping. For training the domain adaptation model, we use 5x2-fold CV due to the small target dataset size. Bayesian Search is implemented with Scikit-Optimize (skopt) to perform hyperparameter search in the inner loop [88]. Figure 4 illustrates our model training workflow.

Table 3 summarises the hyperparameters used in the experimentation. The ranges of hyperparameter searches are chosen conservatively to avoid

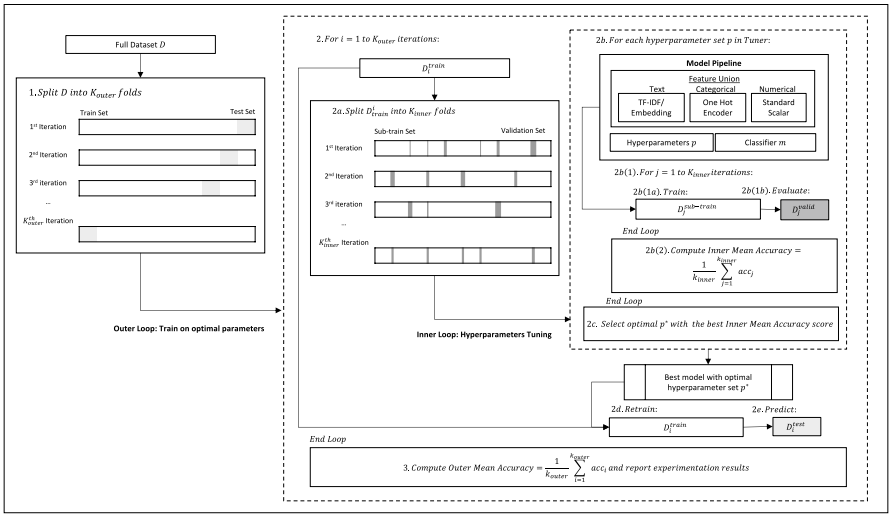

Fig. 4 The nested cross validation (CV) procedure for hyperparameter search and model selection

Table 3 Hyperparameters Setting

Model	Default parameters		Bayesian search parameters	
	Hyperparameters	Values	Hyperparameters	Values
LinearSVC	Class weight	'Balanced'	C	(1e-2, 1e+2)
	Penalty	'l2'		
	Loss	'Squared hinge'		
XGBoost	–	-	Max depth	(3,7)
			Subsample	(0.5, 1)
			Colsample by tree	(0.5, 1)
			Learning rate	(1e-2, 1e-1)
LGBM	Class weight	'balanced'	Max depth	(3, 17)
	Histogram pool size	1024	Num leaves	(32, 128)
			Min child samples	(3, 128)
			Learning rate	(1e-2, 1e-1)
BERT	Batch size	16	–	–
	Num epoch	4		
	Patience	2		
	Learning rate	(3e-5, 0.0)		

out-of-memory issue and long training times, despite a trade-off with model performance. Besides that, the learning rate policy for the transformer models adopts the Polynomial decay scheduler with a warm up at 0.1 timestep to prevent early overfitting [89].

We also observe that the RoBERTa model suffers from catastrophic forgetting when using our default learning rate, as illustrated in Fig. 5. This suggests that curated hyperparameters are necessary to improve the model performance. In our case, competitive performance is achieved when we lower the learning rate to 1e-5.

4.2 Performance metrics

The following metrics are used for evaluation:

- **Accuracy** measures the ratio of correctly predicted labels to the total predicted labels.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

- **F1** measures the harmonic mean of the precision and recall metrics.

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{7}$$

- **ROC AUC** measures the area under the Receiver Operating Characteristic Curve (ROC), which plots the trade-offs between sensitivity against 1-specificity, where TP,TN,FP,FN, are true positive, true negative, false positive, and false negative, respectively.

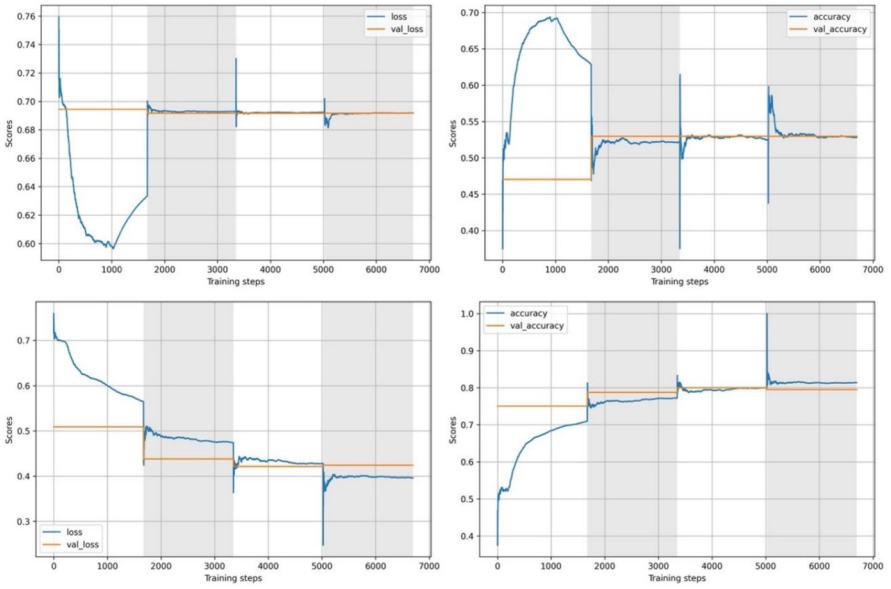

Fig. 5 RoBERTa model exhibits catastrophic forgetting with respect to fine-tuning on the ParlVote-large subset with the default learning rate (top graphs) vs. pptimal performance with a lower learning rate (bottom graphs)

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$1 - Specificity = \frac{FP}{TN + FP} \tag{9}$$

5 Results and discussions

5.1 Cross validation results for benchmarking

Table 4 summarises the experimental results for benchmarking the UK Hansard datasets, focusing on accuracy to compare the performance of machine learning models against previous studies. The results are reported over 10-fold cross validation, with the best scores highlighted in bold. All models outperform the dummy classifier. Besides that, the model improves slightly with the use of higher n-grams and pre-processing of text. After adding handcrafted features, all models significantly outperform the baselines in both t-test and Wilcoxon signed rank test. Further details on model performance of other metrics (F1 and ROC AUC scores), the significance test results, and feature importance are presented in Appendix.

Keeping the input representation fixed with trigram TF-IDF and non-text features, the best model identified is the LGBM classifier, which performs the best for medium to large datasets, compared to the LinearSVC and XGBoost classifiers. A trade-off is observed in model performance, with a significantly high

Table 4 Performance of ML models on stance classification task—accuracy score

Model- random split	HanDeSet	ParlVote				
	Small	Small		Medium		Large
	All	Any	≤ 512	Any	≤ 512	All
Baselines						
Dummy (Majority Class)	0.57	0.53	0.55	0.53	0.55	0.53
Unigram TF-IDF + LinearSVC [14, 23]	0.67	0.62	0.61	0.63	0.61	0.64
Deepwalk [34]	–	0.73	0.73	0.72	0.72	0.72
GPolS [34]	–	**0.80**	**0.80**	0.77	0.77	0.76
Ours						
Text only						
Trigram TF-IDF + LinearSVC	0.69	0.62	0.62	0.64	0.62	0.65
With non-text features						
Trigram TF-IDF + LinearSVC	0.88	**0.80**	0.77	0.78	0.76	0.77
Trigram TF-IDF + XGBoost	**0.90**	**0.80**	0.73	0.79	0.79	0.80
Trigram TF-IDF + LGBM	**0.90**	**0.80**	0.75	0.79	0.80	0.81
Trigram TF-IDF + BERT Features + LGBM	**0.90**	0.79	0.76	**0.80**	**0.81**	**0.82**

Best scores are highlighted in bold text

computational overhead when the number of features increases due to higher n-grams and more complex models, resulting in longer training times. Combining pre-trained BERT embeddings with the LGBM classifier leads to the best performance for medium and large datasets. While increasing the number of training samples can substantially improve model performance, this comes at the cost of even longer training times.

For transformer models, the BERT model fails to learn effectively with small subsets from the ParlVote dataset, while outperforming traditional machine learning models when additional numerical features are absent in larger datasets. This can be attributed to the generalisation ability provided by transfer learning, in contrast to the TF-IDF representation, which is susceptible to the out-of-vocabulary issue owing to its dependency on local training data for feature construction. The BERT-based transformers also outperform the baselines when additional features are included. Overall, traditional machine learning model with TF-IDF text representation can achieve competitive performance compared to transformer models with contextual text representation. We also observed that adding informative features alone could lead to over a 10% improvement in model performance, along with better explainability, whereas the choice of suitable text representation typically only varies model performance by 1-3% after fine-tuning. This suggests that the selection of "optimal" language models may be less critical (Table 5).

The proposed model is further assessed with the group split training strategy. We implement the stratified group k-fold CV strategy for both the inner and outer loops. We use this strategy to prevent data samples from the same debate from appearing in both the training and test sets. This means the model is evaluated on unseen debate topics and vocabulary, while the random split strategy only evaluates the model on unseen samples.

From the results in Table 6, we notice that the random split training strategy can potentially result in overly optimistic performance due to data leakage. While the group split training strategy shows a smaller decrement of approximately 3%-9% accuracy for traditional machine learning models, the accuracy

Table 5 Performance of transformer models on stance classification task—accuracy score

Model- Random split	HanDeSet	ParlVote				
	Small	Small		Medium		Large
	All	Any	≤ 512	Any	≤ 512	All
Text only						
BERT base cased	0.70	0.58	0.56	0.65	0.66	0.68
With non-text features						
BERT base cased	0.88	0.78	0.74	0.78	0.79	0.79
BERT base uncased	0.87	0.78	0.75	0.78	0.80	0.80
DistilBERT base uncased	0.88	0.78	0.75	0.77	0.79	0.79
RoBERTa base cased	0.88	0.78	0.76	0.70	0.73	0.71
RoBERTa base cased - small LR	0.87	0.78	0.76	0.77	0.79	0.80

Table 6 Performance of the stance classification task with group split training—accuracy score

Model- Group split	HanDeSet	ParlVote		
	Small	Small	Medium	Large
Text only				
Dummy (Majority Class)	0.54	0.47	0.53	0.53
Trigram TF-IDF + LinearSVC	0.60	0.58	0.59	0.62
BERT base uncased	0.64	0.55	0.60	0.64
With non-text features				
Trigram TF-IDF + LGBM	**0.81**	**0.77**	0.76	**0.77**
Trigram TF-IDF + BERT Features + LGBM	0.79	0.75	**0.77**	0.76
BERT base uncased	0.67	0.60	0.65	0.65

Best scores are highlighted in bold text

score drops by up to 20% for the transformer model. Besides that, the transformer model requires more training data for model improvement.

5.2 Cross validation results for application

Based on the experimental results, two models with non-text features: Trigram TF-IDF + LGBM and BERT base uncased Transformer, are selected for our subsequent evaluation on the application dataset. Table 7 summarises the experimental results of comparing the differences in model performance before and after applying domain adaptation techniques.

The outcomes show that direct inference on the target dataset using a model trained only on source datasets yields the worst results, while domain adaptation

Table 7 Domain adaptation performance

Method	Accuracy	F1	AUC
HanDeSet only			
Source only	0.69	0.71	0.72
Transfer AdaBoost	0.76	0.76	0.76
Feature augmentation	0.78	0.78	0.78
Parameter fine tune	0.75	0.76	0.83
HanDeSet + ParlVote			
Source only	0.69	0.69	0.73
Transfer AdaBoost	0.77	0.77	0.78
Feature augmentation	**0.79**	**0.79**	**0.80**
Parameter fine tune	0.73	0.74	0.78

Best scores are highlighted in bold text

techniques lead to better model performance, with the best model achieving an accuracy of 0.79 when feature augmentation is applied.

6 Error analysis with case studies

In this section, incorrect predictions on test samples are analysed to better understand the weaknesses of our best-performing model. It is observed that misclassified samples are generally lengthier than correctly classified samples, which implies potential challenges in adequately capturing sequential and contextual information for interpreting the overall stance of a speech. The misclassified samples also tend to have higher sentiment scores (Tables 8, 9 and 10).

Besides that, upon closer examination using the word cloud in Fig. 6, false positive samples appear to contain fewer explicit terms related to stance and are characterised by a lower occurrence of common terms, compared to those from classes that involve discussions on 'health'. We also observe a strong correlation (0.62) between the party affiliation feature and the model's prediction, as shown in Fig. 7.

To gain further insights, two different debate topics from the Australian Hansard are selected for demonstration. We deploy the final best-performing model for error analysis. Since voting did not occur at the time of the debate, we manually annotate the expected labels and underline the phrases that reflect a speaker's stance. Some sentences are truncated for illustration purposes. From the numerical results, the proposed model accurately predicts the statements with stances expressed explicitly. Most misclassified speeches, however, are often not direct responses to a motion but rather responses to another speaker. This implies that concatenating utterances as speeches may not be a good idea. Moreover, we show that the VADER sentiment score and stance do not necessarily align.

Another finding is that the model tends to favour statements from the same party affiliation. For example, Speech 6 in Debate 2 was misclassified as "0" despite the speaker explicitly stating his support for the motion. While it is rare to find conflicting arguments within the same party, we remain cautious when interpreting the prediction output of our stance classifier. We believe that including party affiliation as a feature for stance classification can be a double-edged sword. Despite being an important feature, the assumption on homophily can lead to the potential risk of model bias.

Table 8 Data Summary by Confusion (Median)

Features	TP	TN	FP	FN
Number of words	1090	927	1949	1435
Motion Sentiment	0.05	0.51	0.99	0.76
Speech Sentiment	0.94	0.91	0.99	0.99

Fig. 6 Word cloud on classification samples

7 Conclusions

This research has addressed a gap in the under-studied computational approaches for political text mining on Australian parliamentary debates. A systematic comparative study of stance classification models has been conducted on these debates, leveraging state-of-the-art machine learning and NLP methods. For benchmarking purposes, the random split training strategy was adopted for the first experiment, and the results indicate that, after incorporating the derived features, the performance of all models has improved significantly. In particular, the LGBM classifier performs the best, achieving 90% accuracy on the HanDeSet dataset and 81% on the ParlVote datasets.

This study has also demonstrated that, with a set of representative features, traditional machine learning models can achieve satisfactory performance comparable to transformer models, with the added advantages of shorter training times and better interpretability. Additionally, we have empirically shown that

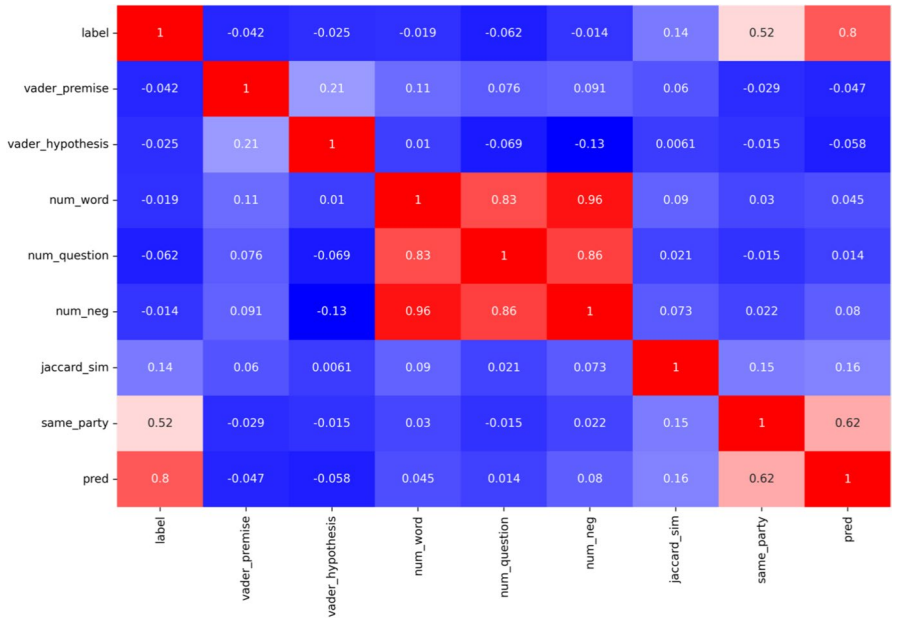

Fig. 7 Feature correlation

Table 9 Case Study 1

Debate 1	Junk Food Advertising Impact on Young Canberrans
Motion	... Control over what advertisements can be shown ...should be implemented
	Party: Labor
Speech 1	This topic is dear to the Greens' heart...I reject utterly Mr Howard's statement that it is nannyism to put a ban on junk food advertising...
	Party: Greens, VADER score: 0.9974, Label: 1, Prediction: 0
Speech 2	... I will be brief in adding a few comments in support of the comments of my colleague Ms Porter and the other contributions ...
	Party: Labor, VADER score: 0. 9565, Label: 1, Prediction: 1
Speech 3	I stand to support the words of my colleagues and thank Ms Porter for raising this matter...
	Party: Greens, VADER score: 0. 9766, Label: 1, Prediction: 1
Speech 4	Point of order. I find this very interesting, but I am wondering what it has to do with food advertising
	Party: Labor, VADER score: 0. 7172, Label: 0, Prediction: 1
Speech 5	...I suspect that whatever is advertised does not have a huge effect on them...
	Party: Liberal, VADER score: 0. 9997, Label: 0, Prediction: 1

Phrases reflecting a speaker's stance are underlined

Table 10 Case Study 2

Debate 2	Obesity and Fast Food
Motion	I move: That this house requests that the Social Development Committee inquire into and report upon the link between obesity and fast foods...
	Party: Labor
Speech 1	The bloke who invented the chicken nugget only recently died
	Party: Labor, VADER score: -0. 5574, Label: 1, Prediction: 1
Speech 2	... Will Macca's do the same? Just in Australia? Only 83 per cent!
	Party: Liberal, VADER score: 0, Label: 0, Prediction: 1
Speech 3	I have put before the house my arguments for supporting this motion...
	Party: Labor, VADER score: 0. 6387, Label: 1, Prediction: 1
Speech 4	Bring back Brindal; bring back someone interesting with a bit of scholarship
	Party: Labor, VADER score: 0. 4019, Label: 0, Prediction: 1
Speech 5	First, I thank the members ...I am opposing this motion...
	Party: Liberal, VADER score: 0. 9797, Label: 0, Prediction: 0
Speech 6	...I support the motion and trust that the reference gives the committee enough scope..
	Party: Indep., VADER score: 0. 9963, Label: 1, Prediction: 0
Speech 7	I rise as a member of the Social Development Committee to support this motion...
	Party: Labor, VADER score: 0. 9892, Label: 1, Prediction: 1

Phrases reflecting a speaker's stance are underlined

the model can perform well without a large training dataset in most scenarios. The developed models were further investigated for cross-topic stance classification using a group split strategy to evaluate generalisability on unseen topics. As expected, the results showed a significant drop in model accuracy for stance classification on unseen topics and vocabulary terms. This outcome implies that the issue of model overfitting could potentially occur in empirical evaluations, yielding unrealistically optimistic results.

We also have tackled the challenge of lacking labelled data by integrating domain adaptation techniques into cross-dataset classification, with the feature augmentation method yielding the best result, achieving 79% accuracy. Finally, error analyses revealed the issue of model bias when using the party affiliation feature for stance classification. This issue will be further investigated in our future work.

A few limitations are identified in the proposed method:

1. The unit of the analysis is speech, which is a concatenation of utterances. This concatenation can cause a loss of context in terms of inter-speaker dependencies, especially since some utterances are not a direct response to the motion. An alternative approach could be to predict the stance for each individual utterance and aggregate the results.
2. Due to constraints on computational resources, a maximum sequence length of 512 tokens is set to truncate text for transformer models in order to reduce com-

putational complexity. Furthermore, hyperparameter searches are not performed for the transformer models, and the range of hyperparameter searches for traditional machine learning models is constrained. These limitations may compromise model performance.
3. More advanced techniques for domain adaptation could be explored, as this study only relies on a few widely used domain adaptation methods.

This study conducted a structured analysis of Australian State and Territory parliamentary debates, highlighting the promising use of machine learning and NLP approaches for stance classification, despite challenges such as data cleaning and the limited availability of labelled data. Looking ahead, further improvements in pre-processing could help address challenges with long text, such as summarising speech openers and thank-you speeches, which are often irrelevant to argumentation and increase computational burden. Incorporating advanced transformer architectures or hierarchical attention mechanisms could also enhance performance. Additionally, exploring other computational techniques in text mining, such as topic modeling and identifying frames of argumentation, could provide a more comprehensive analysis of political texts.

The domain adaptation technique, specifically feature augmentation, was proposed to improve model generalization and reduce overfitting on the source domain. While overfitting may still occur on the target domain, future work could explore strategies such as adding random noise or implementing regularization techniques to mitigate this risk. Instead of relying on traditional feature engineering approaches, future research could consider incorporating pattern recognition, automated machine learning (autoML) techniques, or knowledge-enhanced models. These could potentially improve in-context learning, natural language understanding of argument intention, and detection of implicit stance. Finally, the creation of a new annotated Hansard, including information on dialogue structure, may be needed to facilitate future research on modelling inter-speaker dependencies and using preceding speech to capture contextual information.

7.1 Supplementary information

The source codes and experimentation details will be made available at: https://github.com/stephanienzz/stance-classification.

	Model	HanDeSet	ParlVote				
		Small - 1251	Small - 1251		Medium - 18,253		Large - 33461
		All	Any	<512	Any	<512	All
Baselines	Dummy (Majority Class)	0.73 ± 3e-3	0.69 ± 3e-3	0.71 ± 2e-3	0.69 ± 2e-4	0.71 ± 1e-4	0.69 ± 0e-4
	Unigram TF-IDF + LinearSVC	0.70 ± 2e-2	0.65 ± 5e-2	0.64 ± 5e-2	0.64 ± 1e-2	0.64 ± 1e-2	0.65 ± 7e-3
	Trigram TF-IDF + LinearSVC	0.72 ± 3e-2	0.66 ± 4e-2	0.65 ± 4e-2	0.66 ± 1e-2	0.65 ± 1e-2	0.67 ± 7e-3
	Bert-base-cased	0.74 ± 4e-2	0.64 ± 6e-2	0.61 ± 6e-2	0.66 ± 3e-2	0.69 ± 2e-2	0.70 ± 1e-2
+ Non-text Features	Trigram TF-IDF + LinearSVC	0.89 ± 2e-2	**0.79 ± 5e-2**	**0.78 ± 4e-2**	0.78 ± 6e-3	0.77 ± 6e-3	0.78 ± 9e-3
	Trigram TF-IDF + XGBoost	**0.91 ± 2e-2**	**0.79 ± 3e-2**	0.74 ± 3e-2	0.78 ± 5e-3	0.79 ± 8e-3	0.79 ± 9e-3
	Trigram TF-IDF + LGBM	**0.91 ± 2e-2**	**0.79 ± 4e-2**	0.74 ± 3e-2	0.79 ± 7e-3	0.80 ± 8e-3	0.81 ± 7e-3
	Bert-base-cased	0.89 ± 2e-2	0.78 ± 4e-2	0.75 ± 3e-2	0.78 ± 1e-2	0.80 ± 9e-3	0.80 ± 7e-3
	Bert-base-uncased	0.88 ± 2e-2	0.78 ± 2e-2	0.76 ± 4e-2	0.79 ± 8e-3	**0.81 ± 7e-3**	0.81 ± 7e-3
	DistilBert-base-uncased	0.88 ± 2e-2	0.77 ± 4e-2	0.75 ± 3e-2	0.78 ± 6e-3	0.80 ± 1e-2	0.80 ± 6e-3
	Roberta-base	0.89 ± 2e-2	0.76 ± 4e-2	0.74 ± 3e-2	0.74 ± 3e-2	0.76 ± 4e-2	0.75 ± 4e-2
	Roberta-base-smallLR	0.88 ± 3e-2	0.76 ± 4e-2	0.74 ± 3e-2	0.77 ± 1e-2	0.80 ± 7e-3	0.80 ± 6e-3
	Trigram TF-IDF+BERT+LGBM	**0.91 ± 2e-2**	0.78 ± 4e-2	0.76 ± 2e-2	**0.80 ± 7e-3**	**0.81 ± 8e-3**	**0.82 ± 7e-3**

Fig. 8 Stance classification task performance- F1 score

	Model	HanDeSet	ParlVote				
		Small - 1251	Small - 1251		Medium - 18,253		Large - 33461
		All	Any	<512	Any	<512	All
Baselines	Dummy (Majority Class)	0.50 ± 0e-4	0.50 ± 0e-4	0.50 ± 0e-4	0.50 ± 0e-4	0.50 ± 0e-4	0.50 ± 0e-4
	Unigram TF-IDF + LinearSVC	0.74 ± 3e-2	0.67 ± 4e-2	0.63 ± 6e-2	0.68 ± 2e-2	0.66 ± 1e-2	0.69 ± 6e-3
	Trigram TF-IDF + LinearSVC	0.76 ± 3e-2	0.67 ± 6e-2	0.66 ± 5e-2	0.69 ± 1e-2	0.66 ± 1e-2	0.70 ± 8e-3
	Bert-base-cased	0.77 ± 4e-2	0.59 ± 6e-2	0.60 ± 5e-2	0.71 ± 2e-2	0.71 ± 3e-2	0.75 ± 1e-2
+Non-text Features	Trigram TF-IDF + LinearSVC	0.94 ± 1e-2	0.86 ± 5e-2	**0.84 ± 3e-2**	0.84 ± 7e-3	0.82 ± 6e-3	0.84 ± 9e-3
	Trigram TF-IDF + XGBoost	0.95 ± 1e-2	0.87 ± 3e-2	0.79 ± 3e-2	0.87 ± 7e-3	0.87 ± 8e-3	0.89 ± 7e-3
	Trigram TF-IDF + LGBM	**0.96 ± 1e-2**	0.86 ± 3e-2	0.79 ± 2e-2	0.88 ± 5e-2	0.88 ± 5e-3	**0.90 ± 5e-3**
	Bert-base-cased	0.95 ± 1e-2	0.84 ± 3e-2	0.79 ± 3e-2	0.86 ± 6e-3	0.87 ± 6e-3	0.88 ± 6e-3
	Bert-base-uncased	0.94 ± 2e-2	0.85 ± 2e-2	0.80 ± 4e-2	0.86 ± 5e-3	0.87 ± 1e-2	0.88 ± 6e-3
	DistilBert-base-uncased	0.95 ± 1e-2	0.84 ± 3e-2	0.79 ± 4e-2	0.85 ± 2e-3	0.87 ± 7e-3	0.88 ± 5e-3
	Roberta-base	0.95 ± 2e-2	0.83 ± 3e-2	0.76 ± 3e-2	0.74 ± 2e-1	0.76 ± 1e-1	0.74 ± 2e-1
	Roberta-base-smallLR	0.94 ± 2e-2	0.83 ± 4e-2	0.78 ± 4e-2	0.86 ± 7e-3	0.88 ± 9e-3	0.88 ± 6e-3
	Trigram TF-IDF+BERT+LGBM	**0.96 ± 1e-2**	**0.88 ± 3e-2**	0.82 ± 2e-2	**0.89 ± 7e-2**	**0.89 ± 7e-3**	**0.90 ± 5e-3**

Fig. 9 Stance classification task performance- ROC AUC score

Small any Subset

	model1	model2	m1_accuracy	m2_accuracy	tstat	pvalue	< 0.05?
0	Deepwalk	GPolS	0.73 +/- 4e-3	0.80 +/- 5e-4	-54.912518	0.000000	True
1	Deepwalk	dummy_baseline+DummyClassifier	0.73 +/- 4e-3	0.53 +/- 0.0	123.466200	1.000000	False
2	Deepwalk	tfidf_unigram+LinearSVC	0.73 +/- 4e-3	0.62 +/- 0.04	8.133531	0.999991	False
3	Deepwalk	tfidf_trigram+LinearSVC	0.73 +/- 4e-3	0.62 +/- 0.04	8.083364	0.999991	False
4	Deepwalk	bert-base-cased_baseline+FineTune	0.73 +/- 4e-3	0.58 +/- 0.05	9.845286	0.999998	False
5	Deepwalk	tfidf_features+LinearSVC	0.73 +/- 4e-3	0.8 +/- 0.04	-5.383067	0.000207	True
6	Deepwalk	tfidf_features+XGBClassifier	0.73 +/- 4e-3	0.8 +/- 0.02	-8.467524	0.000005	True
7	Deepwalk	tfidf_features+LGBMClassifier	0.73 +/- 4e-3	0.8 +/- 0.04	-6.052028	0.000086	True
8	Deepwalk	bert-base-cased_features+FineTune	0.73 +/- 4e-3	0.78 +/- 0.03	-4.685444	0.000529	True
9	Deepwalk	bert-base-uncased_features+FineTune	0.73 +/- 4e-3	0.78 +/- 0.02	-8.290932	0.000005	True
10	GPolS	dummy_baseline+DummyClassifier	0.80 +/- 5e-4	0.53 +/- 0.0	263.618582	1.000000	False
11	GPolS	tfidf_unigram+LinearSVC	0.80 +/- 5e-4	0.62 +/- 0.04	13.267570	1.000000	False
12	GPolS	tfidf_trigram+LinearSVC	0.80 +/- 5e-4	0.62 +/- 0.04	13.376768	1.000000	False
13	GPolS	bert-base-cased_baseline+FineTune	0.80 +/- 5e-4	0.58 +/- 0.05	14.451622	1.000000	False
14	GPolS	tfidf_features+LinearSVC	0.80 +/- 5e-4	0.8 +/- 0.04	0.110395	0.542741	False
15	GPolS	tfidf_features+XGBClassifier	0.80 +/- 5e-4	0.8 +/- 0.02	0.384004	0.645058	False
16	GPolS	tfidf_features+LGBMClassifier	0.80 +/- 5e-4	0.8 +/- 0.04	0.197622	0.576135	False
17	GPolS	bert-base-cased_features+FineTune	0.80 +/- 5e-4	0.78 +/- 0.03	1.967840	0.959696	False
18	GPolS	bert-base-uncased_features+FineTune	0.80 +/- 5e-4	0.78 +/- 0.02	3.737400	0.997684	False

Medium any Subset

	model1	model2	m1_accuracy	m2_accuracy	tstat	pvalue	< 0.05?
0	Deepwalk	GPolS	0.72 +/- 9e-4	0.77 +/- 6e-4	-146.176337	0.000000	True
1	Deepwalk	dummy_baseline+DummyClassifier	0.72 +/- 9e-4	0.53 +/- 0.0	653.066612	1.000000	False
2	Deepwalk	tfidf_unigram+LinearSVC	0.72 +/- 9e-4	0.63 +/- 0.01	25.855774	1.000000	False
3	Deepwalk	tfidf_trigram+LinearSVC	0.72 +/- 9e-4	0.64 +/- 0.01	20.076997	1.000000	False
4	Deepwalk	tfidf_features+LinearSVC	0.72 +/- 9e-4	0.78 +/- 0.01	-27.627862	0.000000	True
5	Deepwalk	tfidf_features+XGBClassifier	0.72 +/- 9e-4	0.79 +/- 0.01	-39.848231	0.000000	True
6	Deepwalk	tfidf_features+LGBMClassifier	0.72 +/- 9e-4	0.79 +/- 0.01	-31.681036	0.000000	True
7	GPolS	dummy_baseline+DummyClassifier	0.77 +/- 6e-4	0.53 +/- 0.0	1202.000000	1.000000	False
8	GPolS	tfidf_unigram+LinearSVC	0.77 +/- 6e-4	0.63 +/- 0.01	39.750845	1.000000	False
9	GPolS	tfidf_trigram+LinearSVC	0.77 +/- 6e-4	0.64 +/- 0.01	32.639883	1.000000	False
10	GPolS	tfidf_features+LinearSVC	0.77 +/- 6e-4	0.78 +/- 0.01	-2.798252	0.010212	True
11	GPolS	tfidf_features+XGBClassifier	0.77 +/- 6e-4	0.79 +/- 0.01	-9.968015	0.000002	True
12	GPolS	tfidf_features+LGBMClassifier	0.77 +/- 6e-4	0.79 +/- 0.01	-9.630662	0.000002	True

Fig. 10 *T* test results for external model comparison on ParlVote subsets

Small 512 Subset

	model1	model2	m1_accuracy	m2_accuracy	tstat	pvalue	< 0.05?
0	Deepwalk	GPoIS	0.73 +/- 3e-3	0.80 +/- 4e-4	-73.139216	0.000000	True
1	Deepwalk	dummy_baseline+DummyClassifier	0.73 +/- 3e-3	0.55 +/- 0.0	149.646666	1.000000	False
2	Deepwalk	tfidf_unigram+LinearSVC	0.73 +/- 3e-3	0.61 +/- 0.05	7.419505	0.999981	False
3	Deepwalk	tfidf_trigram+LinearSVC	0.73 +/- 3e-3	0.62 +/- 0.05	7.546575	0.999983	False
4	Deepwalk	bert-base-cased_baseline+FineTune	0.73 +/- 3e-3	0.56 +/- 0.04	14.173401	1.000000	False
5	Deepwalk	tfidf_features+LinearSVC	0.73 +/- 3e-3	0.77 +/- 0.03	-3.813572	0.002001	True
6	Deepwalk	tfidf_features+XGBClassifier	0.73 +/- 3e-3	0.73 +/- 0.03	-0.368061	0.360556	False
7	Deepwalk	tfidf_features+LGBMClassifier	0.73 +/- 3e-3	0.75 +/- 0.03	-2.428856	0.018735	True
8	Deepwalk	bert-base-cased_features+FineTune	0.73 +/- 3e-3	0.74 +/- 0.03	-1.302100	0.112269	False
9	Deepwalk	bert-base-uncased_features+FineTune	0.73 +/- 3e-3	0.75 +/- 0.03	-2.687628	0.012150	True
10	GPoIS	dummy_baseline+DummyClassifier	0.80 +/- 4e-4	0.55 +/- 0.0	318.251463	1.000000	False
11	GPoIS	tfidf_unigram+LinearSVC	0.80 +/- 4e-4	0.61 +/- 0.05	11.763808	1.000000	False
12	GPoIS	tfidf_trigram+LinearSVC	0.80 +/- 4e-4	0.62 +/- 0.05	12.322226	1.000000	False
13	GPoIS	bert-base-cased_baseline+FineTune	0.80 +/- 4e-4	0.56 +/- 0.04	20.057235	1.000000	False
14	GPoIS	tfidf_features+LinearSVC	0.80 +/- 4e-4	0.77 +/- 0.03	3.044194	0.993041	False
15	GPoIS	tfidf_features+XGBClassifier	0.80 +/- 4e-4	0.73 +/- 0.03	8.275263	0.999992	False
16	GPoIS	tfidf_features+LGBMClassifier	0.80 +/- 4e-4	0.75 +/- 0.03	5.547678	0.999822	False
17	GPoIS	bert-base-cased_features+FineTune	0.80 +/- 4e-4	0.74 +/- 0.03	6.457068	0.999942	False
18	GPoIS	bert-base-uncased_features+FineTune	0.80 +/- 4e-4	0.75 +/- 0.03	6.146681	0.999915	False

Medium 512 Subset

	model1	model2	m1_accuracy	m2_accuracy	tstat	pvalue	< 0.05?
0	Deepwalk	GPoIS	0.72 +/- 1e-3	0.77 +/- 5e-4	-141.421356	0.000000	True
1	Deepwalk	dummy_baseline+DummyClassifier	0.72 +/- 1e-3	0.55 +/- 0.0	542.031790	1.000000	False
2	Deepwalk	tfidf_unigram+LinearSVC	0.72 +/- 1e-3	0.61 +/- 0.01	34.100456	1.000000	False
3	Deepwalk	tfidf_trigram+LinearSVC	0.72 +/- 1e-3	0.62 +/- 0.01	29.621492	1.000000	False
4	Deepwalk	tfidf_features+LinearSVC	0.72 +/- 1e-3	0.76 +/- 0.01	-19.185575	0.000000	True
5	Deepwalk	tfidf_features+XGBClassifier	0.72 +/- 1e-3	0.79 +/- 0.01	-26.591804	0.000000	True
6	Deepwalk	tfidf_features+LGBMClassifier	0.72 +/- 1e-3	0.8 +/- 0.01	-30.585620	0.000000	True
7	GPoIS	dummy_baseline+DummyClassifier	0.77 +/- 5e-4	0.55 +/- 0.0	1320.071053	1.000000	False
8	GPoIS	tfidf_unigram+LinearSVC	0.77 +/- 5e-4	0.61 +/- 0.01	50.181119	1.000000	False
9	GPoIS	tfidf_trigram+LinearSVC	0.77 +/- 5e-4	0.62 +/- 0.01	44.763269	1.000000	False
10	GPoIS	tfidf_features+LinearSVC	0.77 +/- 5e-4	0.76 +/- 0.01	5.270872	0.999753	False
11	GPoIS	tfidf_features+XGBClassifier	0.77 +/- 5e-4	0.79 +/- 0.01	-7.720250	0.000014	True
12	GPoIS	tfidf_features+LGBMClassifier	0.77 +/- 5e-4	0.8 +/- 0.01	-10.286540	0.000001	True

Fig. 10 (continued)

Large all Subset

	model1	model2	m1_accuracy	m2_accuracy	tstat	pvalue	< 0.05?
0	Deepwalk	GPolS	0.72 +/- 8e-4	0.76 +/- 3e-4	-148.046642	0.000000	True
1	Deepwalk	dummy_baseline+DummyClassifier	0.72 +/- 8e-4	0.53 +/- 0.0	752.622083	1.000000	False
2	Deepwalk	tfidf_unigram+LinearSVC	0.72 +/- 8e-4	0.64 +/- 0.01	38.382661	1.000000	False
3	Deepwalk	tfidf_trigram+LinearSVC	0.72 +/- 8e-4	0.65 +/- 0.01	28.069993	1.000000	False
4	Deepwalk	tfidf_features+LinearSVC	0.72 +/- 8e-4	0.77 +/- 0.01	-19.816174	0.000000	True
5	Deepwalk	tfidf_features+XGBClassifier	0.72 +/- 8e-4	0.8 +/- 0.01	-30.904319	0.000000	True
6	Deepwalk	tfidf_features+LGBMClassifier	0.72 +/- 8e-4	0.81 +/- 0.01	-42.951330	0.000000	True
7	GPolS	dummy_baseline+DummyClassifier	0.76 +/- 3e-4	0.53 +/- 0.0	2428.629243	1.000000	False
8	GPolS	tfidf_unigram+LinearSVC	0.76 +/- 3e-4	0.64 +/- 0.01	57.476985	1.000000	False
9	GPolS	tfidf_trigram+LinearSVC	0.76 +/- 3e-4	0.65 +/- 0.01	44.401226	1.000000	False
10	GPolS	tfidf_features+LinearSVC	0.76 +/- 3e-4	0.77 +/- 0.01	-5.019316	0.000357	True
11	GPolS	tfidf_features+XGBClassifier	0.76 +/- 3e-4	0.8 +/- 0.01	-15.040000	0.000000	True
12	GPolS	tfidf_features+LGBMClassifier	0.76 +/- 3e-4	0.81 +/- 0.01	-24.075530	0.000000	True

Fig. 10 (continued)

	model1	model2	m1_accuracy	m2_accuracy	tstat	pvalue	< 0.05?
0	dummy_baseline+DummyClassifier	tfidf_unigram+LinearSVC	0.57 +/- 0.0	0.67 +/- 0.03	0.000000	0.000977	True
1	dummy_baseline+DummyClassifier	tfidf_trigram+LinearSVC	0.57 +/- 0.0	0.69 +/- 0.03	0.000000	0.000977	True
2	dummy_baseline+DummyClassifier	bert-base-cased_baseline+FineTune	0.57 +/- 0.0	0.7 +/- 0.04	0.000000	0.000977	True
3	dummy_baseline+DummyClassifier	tfidf_features+LinearSVC	0.57 +/- 0.0	0.88 +/- 0.03	0.000000	0.000977	True
4	dummy_baseline+DummyClassifier	tfidf_features+XGBClassifier	0.57 +/- 0.0	0.9 +/- 0.03	0.000000	0.000977	True
5	dummy_baseline+DummyClassifier	tfidf_features+LGBMClassifier	0.57 +/- 0.0	0.9 +/- 0.02	0.000000	0.000977	True
6	dummy_baseline+DummyClassifier	bert-base-cased_features+FineTune	0.57 +/- 0.0	0.88 +/- 0.02	0.000000	0.000977	True
7	dummy_baseline+DummyClassifier	bert-base-uncased_features+FineTune	0.57 +/- 0.0	0.87 +/- 0.02	0.000000	0.000977	True
8	tfidf_unigram+LinearSVC	tfidf_trigram+LinearSVC	0.67 +/- 0.03	0.69 +/- 0.03	7.500000	0.018555	True
9	tfidf_unigram+LinearSVC	bert-base-cased_baseline+FineTune	0.67 +/- 0.03	0.7 +/- 0.04	7.500000	0.018555	True
10	tfidf_unigram+LinearSVC	tfidf_features+LinearSVC	0.67 +/- 0.03	0.88 +/- 0.03	0.000000	0.000977	True
11	tfidf_unigram+LinearSVC	tfidf_features+XGBClassifier	0.67 +/- 0.03	0.9 +/- 0.03	0.000000	0.000977	True
12	tfidf_unigram+LinearSVC	tfidf_features+LGBMClassifier	0.67 +/- 0.03	0.9 +/- 0.02	0.000000	0.000977	True
13	tfidf_unigram+LinearSVC	bert-base-cased_features+FineTune	0.67 +/- 0.03	0.88 +/- 0.02	0.000000	0.000977	True
14	tfidf_unigram+LinearSVC	bert-base-uncased_features+FineTune	0.67 +/- 0.03	0.87 +/- 0.02	0.000000	0.000977	True
15	tfidf_trigram+LinearSVC	bert-base-cased_baseline+FineTune	0.69 +/- 0.03	0.7 +/- 0.04	17.000000	0.161133	False
16	tfidf_trigram+LinearSVC	tfidf_features+LinearSVC	0.69 +/- 0.03	0.88 +/- 0.03	0.000000	0.000977	True
17	tfidf_trigram+LinearSVC	tfidf_features+XGBClassifier	0.69 +/- 0.03	0.9 +/- 0.03	0.000000	0.000977	True
18	tfidf_trigram+LinearSVC	tfidf_features+LGBMClassifier	0.69 +/- 0.03	0.9 +/- 0.02	0.000000	0.000977	True
19	tfidf_trigram+LinearSVC	bert-base-cased_features+FineTune	0.69 +/- 0.03	0.88 +/- 0.02	0.000000	0.000977	True
20	tfidf_trigram+LinearSVC	bert-base-uncased_features+FineTune	0.69 +/- 0.03	0.87 +/- 0.02	0.000000	0.000977	True
21	bert-base-cased_baseline+FineTune	tfidf_features+LinearSVC	0.7 +/- 0.04	0.88 +/- 0.03	0.000000	0.000977	True
22	bert-base-cased_baseline+FineTune	tfidf_features+XGBClassifier	0.7 +/- 0.04	0.9 +/- 0.03	0.000000	0.000977	True
23	bert-base-cased_baseline+FineTune	tfidf_features+LGBMClassifier	0.7 +/- 0.04	0.9 +/- 0.02	0.000000	0.000977	True
24	bert-base-cased_baseline+FineTune	bert-base-cased_features+FineTune	0.7 +/- 0.04	0.88 +/- 0.02	0.000000	0.000977	True
25	bert-base-cased_baseline+FineTune	bert-base-uncased_features+FineTune	0.7 +/- 0.04	0.87 +/- 0.02	0.000000	0.000977	True
26	tfidf_features+LinearSVC	tfidf_features+XGBClassifier	0.88 +/- 0.03	0.9 +/- 0.03	9.500000	0.032227	True
27	tfidf_features+LinearSVC	tfidf_features+LGBMClassifier	0.88 +/- 0.03	0.9 +/- 0.02	8.500000	0.047867	True
28	tfidf_features+LinearSVC	bert-base-cased_features+FineTune	0.88 +/- 0.03	0.88 +/- 0.02	18.000000	0.295505	False
29	tfidf_features+LinearSVC	bert-base-uncased_features+FineTune	0.88 +/- 0.03	0.87 +/- 0.02	34.000000	0.915549	False
30	tfidf_features+XGBClassifier	tfidf_features+LGBMClassifier	0.9 +/- 0.03	0.9 +/- 0.02	19.500000	0.584206	False
31	tfidf_features+XGBClassifier	bert-base-cased_features+FineTune	0.9 +/- 0.03	0.88 +/- 0.02	45.000000	0.967773	False
32	tfidf_features+XGBClassifier	bert-base-uncased_features+FineTune	0.9 +/- 0.03	0.87 +/- 0.02	54.000000	0.999023	False
33	tfidf_features+LGBMClassifier	bert-base-cased_features+FineTune	0.9 +/- 0.02	0.88 +/- 0.02	43.000000	0.947266	False
34	tfidf_features+LGBMClassifier	bert-base-uncased_features+FineTune	0.9 +/- 0.02	0.87 +/- 0.02	36.000000	0.994243	False
35	bert-base-cased_features+FineTune	bert-base-uncased_features+FineTune	0.88 +/- 0.02	0.87 +/- 0.02	29.000000	0.938487	False

Fig. 11 Wilcoxon test results for internal model comparison on HanDeSet

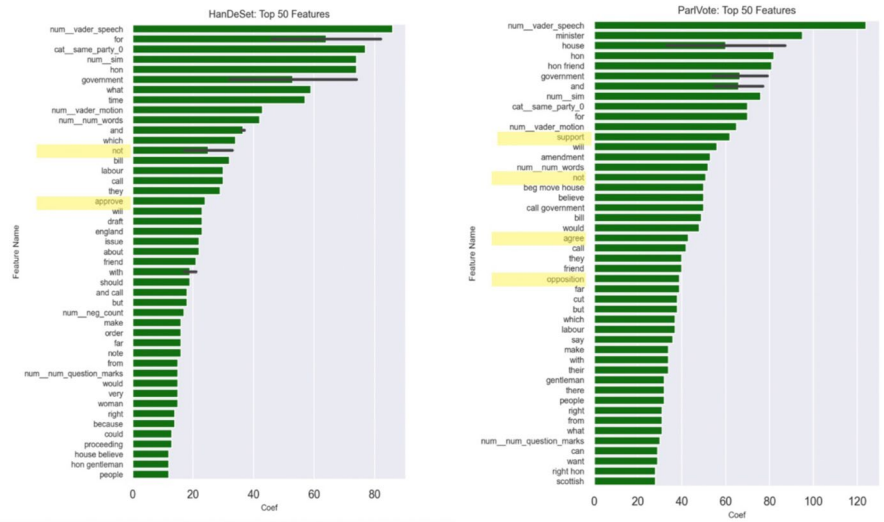

Fig. 12 Feature importance of trigram TF-IDF + LGBM model

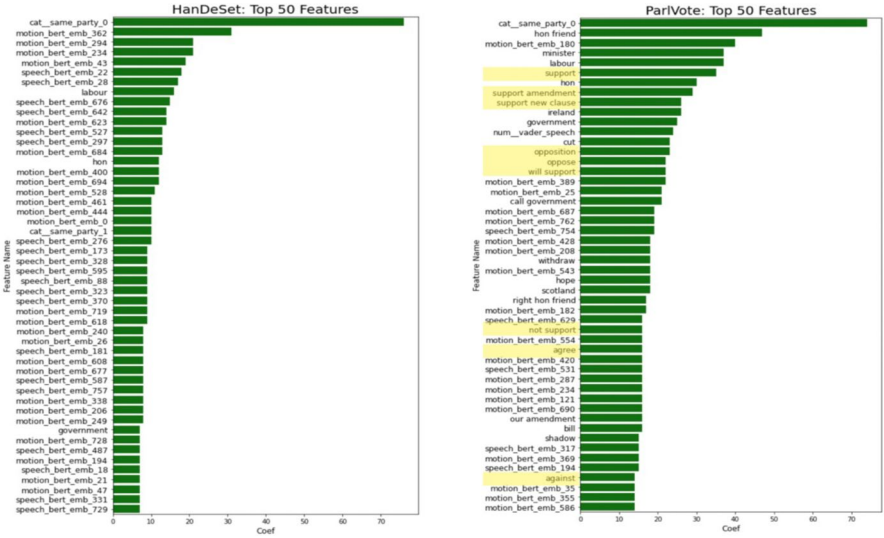

Fig. 13 Feature importance feature importance of trigram TF-IDF + BERT + LGBM model

Appendix A: supplementary results

Figures 8 and 9 summarise the model performance results using the F1 score and ROC AUC metrics.

Figures 10 and 11 summarise the significant test results for the models we experimented with. We present t-test for pairwise model comparisons with external results (since we only have the summary statistics of the published results) and the Wilcoxon signed-rank test for internal model comparisons.

Figures 12 and 13 show the feature importance pertaining to the LGBM classifier based on the HanDeSet and ParlVote datasets, using the best cross-validation results. Higher weights are assigned to numerical features for predicting a speaker's stance. Besides that, keywords indicative of the speaker's stance, such as "agree", "approve", "not", "opposition", and "support" are automatically recognised as important features. For both datasets, the VADER sentiment score for speech is identified as the most important feature. However, when BERT embedding are included, higher weights are given to the $Same_Party$ feature, which reflects the homophily behaviour of speakers. As a result, the model performs well without learning the specific pro and con vocabulary related to a debate topic. Nonetheless, the features derived from BERT-based models are challenging to interpret.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The data that support the findings of this study can be accessed at IEEE Dataport [61].

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval Our research is focused on political text scrutiny and discourse evaluation on publicly available data using computational approaches. While our devised approach could efficiently aid researchers to gain valuable insights in political text analysis, it is important to note potential biases inherent in both training data and model itself, which could lead to a bias evaluation of political discourse. One notable limitation of our approach is the unintended truncation of dialogues or utterances, which could result in potential loss of context and meaning of the original messages. We also acknowledge that these models could potentially influence public opinion and shape political narratives, which could result in unintended consequences such as the spread of misinformation when the results are misinterpreted. As such, it is necessary to conduct further research for addressing these ethical considerations and improving model transparency and interpretability.

References

1. Matthes, J., & Kohring, M. (2008). The content analysis of media frames: toward improving reliability and validity. *Journal of Communication, 58*(2), 258–279. https://doi.org/10.1111/j.1460-2466.2008.00384.x
2. Grimmer, J., & Stewart, B. M. (2013). Text as data: the promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis, 21*(3), 267–297. https://doi.org/10.1093/pan/mps028
3. Nicholls, T., & Culpepper, P. D. (2021). Computational identification of media frames: strengths, weaknesses, and opportunities. *Political Communication, 38*(1–2), 159–181. https://doi.org/10.1080/10584609.2020.1812777
4. Doan, T. M., & Gulla, J. A. (2022). A survey on political viewpoints identification. *Online Social Networks and Media*. https://doi.org/10.1016/j.osnem.2022.100208
5. Thomas, M., Pang, B., & Lee, L. (2006). Get Out the Vote: Determining Support or Opposition from Congressional Floor-debate Transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney.
6. Somasundaran, S., & Wiebe, J. (2010). Recognizing Stances in Ideological Online Debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, LA.
7. Mohammad, S., Kiritchenko, S., P. Sobhani, X. Z., & Cherry, C. (2016). Semeval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego.
8. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: a data mining perspective. *ACM SIGKDD Explorations Newsletter, 19*(1), 22–36. https://doi.org/10.1145/3137597.3137600
9. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: a survey. *ACM Computing Surveys, 51*(2), 1–36. https://doi.org/10.1145/3161603
10. Küçük, D., & Can, F. (2020). Stance detection: a survey. *ACM Computing Survey, 53*(1), 1–37. https://doi.org/10.1145/3369026
11. Beelen, K., Thijm, T. A., Cochrane, C., Halvemaan, K., Hirst, G., Kimmins, M., Lijbrink, S., Marx, M., Naderi, N., Rheault, L., Polyanovsky, R., & Whyte, T. (2017). Digitization of the Canadian parliamentary debates. *Canadian Journal of Political Science, 50*(3), 849–864.
12. Rheault, L., & Cochrane, C. (2020). Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis, 28*(1), 112–133. https://doi.org/10.1017/pan.2019.26
13. Duthie, R., Budzynska, K., & Reed, C. (2016). Mining Ethos in Political Debate. In *Proceedings from the Sixth International Conference on Computational Models of Argument (COMMA)*, pp. 299-310. IOS Press, Netherlands.
14. Abercrombie, G., & Batista-Navarro, R. (2018). Aye' or 'No'? Speech-level Sentiment Analysis of Hansard UK Parliamentary Debate Transcripts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki.
15. Vilares, D., & He, Y. (2017). Detecting Perspectives in Political Debates. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen.
16. Slapin, J. B., Kirkland, J. H., Lazzaro, J. A., Leslie, P. A., & O'Grady, T. (2018). Ideology, grandstanding, and strategic party disloyalty in the British parliament. *American Political Science Review, 112*(1), 15–30. https://doi.org/10.1017/S0003055417000375
17. Fišer, D., & Lenardic, J. (2017). Parliamentary Corpora in the CLARIN infrastructure. In *Selected papers from the CLARIN Annual Conference*, pp. 75-85., Budapest.
18. Lewis, J. M., & Turpin, A. (2017). Understanding the Policy Process Over Time: Linking Debates to Decisions Through Digital Sources. In *Proceedings of International Conference on Public Policy 2017*, Singapore.
19. Abercrombie, G., & Batista-Navarro, R. (2020). Sentiment and position-taking analysis of parliamentary debates: A systematic literature review. *Journal of Computational Social Science, 3*, 245–270. https://doi.org/10.1007/s42001-019-00060-w
20. Mohammad, S., Sobhani, P., & Kiritchenko, S. (2017). Stance and sentiment in tweets. *ACM Transactions on Internet Technology, 17*(3), 1–23. https://doi.org/10.1145/3003433

21. Bestvater, S., & Monroe, B. (2023). Sentiment is not stance: Target-aware opinion classification for political text analysis. *Political Analysis, 31*(2), 235–256. https://doi.org/10.1017/pan.2022.10
22. Fu, Y., Li, X., Li, Y., Wang, S., Li, D., Liao, J., & Zheng, J. (2022). Incorporate opinion-towards for stance detection. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2022.108657
23. Abercrombie, G., & Batista-Navarro, R. (2020). ParlVote: A Corpus for Sentiment Analysis of Political Debates. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 5073-5078. European Language Resources Association, Marseille.
24. Bhavan, A., Mishra, R., Sinha, P. P., Sawhney, R., & Shah, R. (2019). Investigating Political Herd Mentality: A Community Sentiment Based Approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Italy.
25. Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S.-G., Almerekhi, H., & Jansen, B. J. (2020). Developing an online hate classifier for multiple social media platforms. *Human-Centric Computing and Information Sciences*. https://doi.org/10.1186/s13673-019-0205-6
26. He, L., Wang, X., Chen, H., & Xu, G. (2022). Online spam review detection: a survey of literature. *Human-Centric Intelligent Systems, 2*(1–2), 14–30. https://doi.org/10.1007/s44230-022-00001-3
27. Hasan, M. M., Zaman, S. M., Talukdar, M. A., Siddika, A., & Alam, M. G. R. (2021). An Analysis of Machine Learning Algorithms and Deep Neural Networks for Email Spam Classification using Natural Language Processing. *In Proceedings of 2021 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 1-6. IEEE.
28. Alzamzami, F., Hoda, M., & Saddik, A. E. (2020). Light gradient boosting machine for general sentiment classification on short texts: a comparative evaluation. *IEEE Access, 8*, 101840–101858. https://doi.org/10.1109/ACCESS.2020.2997330
29. Athanasiou, V., & Maragoudakis, M. (2017). A novel, gradient boosting framework for sentiment analysis in languages where nlp resources are not plentiful: a case study for modern Greek. *Algorithms, 10*(1), 34. https://doi.org/10.3390/a10010034
30. Ghadi, A. S., & Fennan, A. (2023). Enhanced sentiment analysis based on improved word embeddings and xgboost. *International Journal of Electrical and Computer Engineering, 13*(2), 1827–1836. https://doi.org/10.11591/ijece.v13i2.pp1827-1836
31. Qi, Z. (2020). The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model. In *Proceedings of IEEE International Conference on Artificial Intelligence and Computer Applications*, Dalian, https://doi.org/10.1109/ICAICA50127.2020.9182555.
32. Haumahu, J. P., Permana, S. D. H., & Yaddarabullah, Y. (2021). Fake news classification for Indonesian news using Extreme Gradient Boosting (XGBoost). In *Proceedings of IOP Conference Series: Materials Science and Engineering*. IOP Publishing.
33. Hussain, A. S. K., Asghar, M. Z., Saddozai, F. K., Arif, A., & Khalid, H. A. (2020). Personality classification from online text using machine learning approach. *International Journal of Advanced Computer Science and Applications (IJACSA)*. https://doi.org/10.14569/IJACSA.2020.0110358
34. Wadhwa, R. S., Agarwal, S., & Shah, R. R. (2020). GPolS: A Contextual Graph-Based Language Model for Analyzing Parliamentary Debates and Political Cohesion. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4847-4859. International Committee on Computational Linguistics, Barcelona.
35. Hasan, K. S., & Ng, V. (2013). Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 1348-1356. Asian Federation of Natural Language Processing, Nagoya.
36. Anand, P., Walker, M., Abbott, R., Tree, J. E. F., Bowmani, R., & Minor, M. (2011). Cats Rule and Dogs Drool!: Classifying Stance in Online Debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, Portland.
37. Hou, W., Li, Y., Liu, Y., & Li, Q. (2022). Leveraging multidimensional features for policy opinion sentiment prediction. *Information Sciences, 610*, 215–234. https://doi.org/10.1016/j.ins.2022.08.004
38. Sirrianni, J. W., Liu, X., & Adams, D. (2021). Predicting stance polarity and intensity in cyber argumentation with deep bidirectional transformers. *IEEE Transactions on Computational Social Systems, 8*(3), 655–667. https://doi.org/10.1109/TCSS.2021.3056596
39. Darwish, K., Magdy, W., & Zanouda, T. (2017). Improved Stance Prediction in a User Similarity Feature Space. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Sydney, https://doi.org/10.1145/3110025.3110112.

40. Rill, S., Reinel, D., Scheidt, J., & Zicari, R. V. (2014). Politwi: Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis. *Knowledge-Based Systems, 69*, 24–33. https://doi.org/10.1016/j.knosys.2014.05.008
41. Bar-Haim, R., Bhattacharya, I., Dinuzzo, F., Saha, A., & Slonim, N. (2017). Stance Classification of Context-Dependent Claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Long Papers, Valencia.
42. Körner, E., Wiedemann, G., Hakimi, A.D., Heyer, G., & Potthast, M. (2021). On Classifying whether Two Texts are on the Same Side of an Argument. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic.
43. Ghosh, S., Singhania, P., Singh, S., Rudra, K., & Ghosh, S. (2019). Stance Detection in Web and Social Media: A Comparative Study. In *Proceedings of Experimental IR Meets Multilinguality, Multimodality, and Interaction*. CLEF 2019. Lugano, https://doi.org/10.1007/978-3-030-28577-7_4.
44. Gera, P., & Neal, T. (2022). A Comparative Analysis of Stance Detection Approaches and Datasets. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*.
45. Schiller, B., Daxenberger, J., & Gurevych, I. (2021). Stance detection benchmark: How robust is your stance detection? *KI-Künstliche Intelligenz, 35*, 329–341. https://doi.org/10.1007/s13218-021-00714-w
46. Ng, L., & Carley, K. (2022). Is my stance the same as your stance? a cross validation study of stance detection datasets. *Information Processing & Management*. https://doi.org/10.1016/j.ipm.2022.103070
47. Sobhani, P., Inkpen, D., & Zhu, X. (2017). A Dataset for Multi-target Stance Detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Short Papers, Valencia.
48. Xu, C., Paris, C., Nepal, S., & Sparks, R. (2018). Cross-target Stance Classification with Self-attention Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne.
49. Khiabani, P. J., & Zubiaga, A. (2023). Few-shot learning for cross-target stance detection by aggregating multimodal embeddings. *IEEE Transactions on Computational Social Systems*. https://doi.org/10.1109/TCSS.2023.3264114
50. Pavan, M. C., & Paraboni, I. (2022). Cross-target Stance Classification as Domain Adaptation. In O. Pichardo Lagunas, J. Martínez-Miranda, & B. Martínez Seis (Eds.), *Advances in Computational Intelligence: 21st Mexican International Conference on Artificial Intelligence, MICAI 2022, Monterrey, Mexico, October 24–29, 2022, Proceedings, Part I* (pp. 15–25). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-19493-1_2
51. Rietzler, A., Stabinger, S., Opitz, P., & Engl, S. (2019). Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France. pp. 4933-4941. European Language Resources Association (ELRA).
52. Sang, E., Schraagen, M., Wang, S., & Dastani, M. (2021). Transfer Learning for Stance Analysis in COVID-19 Tweets. In *Proceedings of CLIN31: Computational Linguistics in The Netherlands*.
53. Daume-III, H. (2007). Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague.
54. W. Dai, G.X. Q. Yang, Yu, Y. (2007). Boosting for Transfer Learning. In *Proceedings of ICML*.
55. Huang, X., Rao, Y., Xie, H., Wong, T.-L., & Wang, F. L. (2017). Cross-domain Sentiment Classification via Topic-related TrAdaBoost. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco.
56. Zheng, L., Liu, G., Yan, C., Jiang, C., Zhou, M., & Li, M. (2020). Improved tradaboost and its application to transaction fraud detection. *IEEE Transactions on Computational Social Systems, 7*(5), 1304–1316. https://doi.org/10.1109/TCSS.2020.3017013
57. Da, W., Jin, O., Xue, G. R., Yang, Q., & Yu, Y. (2009). Eigentransfer: a unified framework for transfer learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 193-200).
58. Boireau, M. (2014). Determining Political Stances from Twitter Timelines: The Belgian Parliament Case. In *Proceedings of the 2014 Conference on Electronic Governance and Open Society: Challenges in Eurasia*, St. Petersburg Russian Federation. https://doi.org/10.1145/2729104.2729114.

59. Lai, M., Cignarella, A. T., Farías, D. I. H., Bosco, C., Patti, V., & Rosso, P. (2020). Multilingual stance detection in social media political debates. *Computer Speech & Language*. https://doi.org/10.1016/j.csl.2020.101075
60. Haq, E. U., Braud, T., Kwon, Y. D., & Hui, P. (2020). A survey on computational politics. *IEEE Access, 8*, 197379–197406. https://doi.org/10.1109/ACCESS.2020.3034983
61. Ng, S. (2023). Hansard. IEEE Dataport https://doi.org/10.21227/53t5-fw17.
62. Joachims, T. (1997). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In *Proceedings of ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, San Francisco.
63. Cahyani, D. E., & Patasik, I. (2021). Performance comparison of tf-idf and word2vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics, 10*(5), 2780–2788.
64. Abubakar, H. D., Umar, M., & Bakale, M. A. (2022). Sentiment classification: review of text vectorization methods: Bag of words, tf-idf, word2vec and doc2vec. *SLU Journal of Science and Technology, 4*(1), 27–33.
65. Meijer, H., Truong, J., & Karimi, R. (2021). Document embedding for scientific articles: Efficacy of word embeddings vs tfidf. arXiv preprint arXiv:2107.05151.
66. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT, Minneapolis, Minnesota*. pp. 4171-4186. Association for Computational Linguistics.
67. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. In *Proceedings of 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing—NeurIPS 2019*, https://doi.org/10.48550/arXiv.1910.01108.
68. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. Preprint at https://doi.org/10.48550/arXiv.1907.11692.
69. Chae, Y., & Davidson, T. (2023). *Large language models for text classification: From zero-shot learning to fine-tuning*. Open Science Foundation.
70. Cruickshank, I. J., & Ng, L. H. X. (2023). Use of large language models for stance classification. Preprint at arXiv preprint arXiv:2309.13734.
71. Zhang, B., Ding, D., & Jing, L. (2022). How would stance detection techniques evolve after the launch of chatgpt? Preprint at arXiv preprint arXiv:2212.14548.
72. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., & Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Preprint at https://doi.org/10.48550/arXiv.1609.08144.
73. Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Proceedings of the International AAAI Conference on Web and Social Media*, Ann Arbor, https://doi.org/10.1609/icwsm.v8i1.14550.
74. Kikteva, Z., Gorska, K., Siskou, W., Hautli, A., & Reed, C. (2022). The keystone role played by questions in debate. In: *Proceedings of the 3rd Workshop on Computational Approaches to Discourse*, pp. 54–63.
75. Hautli-Janisz, A., Budzynska, K., McKillop, C., Plüss, B., Gold, V., & Reed, C. (2022). Questions in argumentative dialogue. *Journal of Pragmatics, 188*, 56–79.
76. Boucek, F. (2002). The structure and dynamics of intra-party politics in Europe. *Perspectives on European Politics and Society, 3*(3), 453–493. https://doi.org/10.1080/15705850208438845
77. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*(85), 2825–2830.
78. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., & Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online.
79. Abadi, M., Agarwal, Ashish, Barham, P., Brevdo, E., Chen, Z., Citro, C., & Zheng, X. (2015). TensorFlow: Large-scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org
80. Suthaharan, S. (2016). *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*. Boston, MA: Springer US.
81. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232.

82. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, https://doi.org/10.1145/2939672.2939785.
83. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach California.
84. Maaten, L. V., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research, 9*(86), 2579–2605.
85. Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11*, 2079–2107.
86. Wainer, J., & Cawley, G. (2021). Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2021.115222
87. O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., & Invernizzi, L. (2019). KerasTuner. Software available from https://github.com/keras-team/keras-tuner.
88. Head, T., MechCoder, Luoppe, G., Shcherbatyi, I., & Fabisch, A. (2018). scikit-optimize/scikit-optimize: v0.5.2. Software available from Zenodo. https://doi.org/10.5281/zenodo.1207017.
89. Mishra, P., & Sarawadekar, K. (2019). Polynomial Learning Rate Policy with Warm Restart for Deep Neural Network. In *Proceedings of TENCON 2019–2019 IEEE Region 10 Conference*, Kochi, https://doi.org/10.1109/TENCON.2019.8929465.

Authors and Affiliations

Stephanie Ng[1] · James Zhang[1] · Samson Yu[2] · Asim Bhatti[1] · Kathryn Backholer[3] · C. P. Lim[1]

✉ Stephanie Ng
szng@deakin.edu.au

James Zhang
james.z@deakin.edu.au

Samson Yu
s.yu@ieee.org

Asim Bhatti
asim.bhatti@deakin.edu.au

Kathryn Backholer
kathryn.backholer@deakin.edu.au

C. P. Lim
chee.lim@deakin.edu.au

[1] Institute for Intelligent Systems Research and Innovation, Deakin University, 75 Pigdons Rd, Waurn Ponds, VIC 3216, Australia

[2] School of Engineering, Deakin University, 75 Pigdons Rd, Waurn Ponds, VIC 3216, Australia

[3] Global Centre for Preventive Health and Nutrition, Institute for Health Transformation, Deakin University, 75 Pigdons Rd, Waurn Ponds, VIC 3216, Australia