DEBACER: a method for slicing moderated debates

Thomas Palmeira Ferraz[1], Alexandre Alcoforado[1], Enzo Bustos[1], André Seidel Oliveira[1], Rodrigo Gerber[1], Naíde Müller[3], André Corrêa d'Almeida[4], Bruno Miguel Veloso[2], Anna Helena Reali Costa[1]

[1]Escola Politécnica, Universidade de São Paulo (USP), São Paulo, SP, Brazil

[2]Universidade Portucalense & INESC TEC, Porto, Portugal

[3]Catholic University of Portugal, Lisboa, Portugal

[4]Columbia University, New York, NY, USA

{thomas.ferraz,alexandre.alcoforado,enzobustos}@usp.br,

{rodrigo.gerber,andre.seidel,anna.reali}@usp.br,

bruno.m.veloso@inesctec.pt, ac3133@sipa.columbia.edu

Abstract. *Subjects change frequently in moderated debates with several participants, such as in parliamentary sessions, electoral debates, and trials. Partitioning a debate into blocks with the same subject is essential for understanding. Often a moderator is responsible for defining when a new block begins so that the task of automatically partitioning a moderated debate can focus solely on the moderator's behavior. In this paper, we (i) propose a new algorithm,* DEBACER*, which partitions moderated debates; (ii) carry out a comparative study between conventional and BERTimbau pipelines; and (iii) validate* DEBACER *applying it to the minutes of the Assembly of the Republic of Portugal. Our results show the effectiveness of* DEBACER*.*

Keywords. *Natural Language Processing, Political Documents, Spoken Text Processing, Speech Split, Dialogue Partitioning*

1. Introduction

During a long debate with numerous participants, it is usual to have several subject changes. Partitioning these dialogues into *blocks of speeches* that represent different stages of the debate is essential to understand what is being said by each participant and to perform various natural language processing tasks. This partitioning of dialogues can be beneficial when one wants to analyze and evaluate the debate, *e.g.*, tracking the evolution of an organization's meetings over several years, indicating how many times a member of the organization has participated in discussions about a particular subject of interest. Thus, this dialogue partitioning procedure can benefit sentiment analysis, position-taking detection, argument agreements, and topic modeling tasks.

Current dialogue state tracking approaches model the entire dialogue as an input graph [Ales et al. 2018], or even model the dialogue as a sequential problem, using techniques such as recurrent neural networks [Wen et al. 2017] and attention models [Shan et al. 2020] to capture the state of the dialog. While these methods can be very efficient in real-time applications that assist other agents in taking actions, such as in

conversational AI, they do not address the existence of multiple participants in the conversation, they are too complex for just partitioning the dialogues into blocks of speech, especially when the characteristics of those dialogues can guide such partitioning.

This is the case of *moderated debates*, forums in which the discussion is organized and controlled by a person in charge of giving the floor to each of the participants in turn. An excellent example of moderated debate is a parliamentary session, where the representatives participate in public hearings and formulate new laws. The documents produced by these discussions are extensive and complex for an ordinary citizen to understand and search for specific information. Electoral debates, public hearings, and trials are other examples of dialogues organized with a moderator. We here argue that the moderator plays an essential role in these discussions, defining the maintenance of the current block of speeches or the beginning of a new one. In this case, it is possible to transform this complex problem into a simpler one that does not require processing the entire history of the dialogue or even the use of complex representations such as graphs.

In this work, we contribute a new algorithm, `DEBACER`, that uses a machine-learning classification pipeline to slice moderated debates, helping to extract insightful information from their transcripts. We validate this method in a case of moderated debates in Portuguese: the Assembly of the Republic of Portugal. We assess the opening moments of sessions, when participants can bring new topics or respond to the previous speaker. In these moments, the chairperson acts as a moderator, managing the entire dialogue: interrupts a very long speech, allows retorts or complements to a statement, or even gives the floor to another interlocutor on a new subject. We also make a comparative study of what is the best classification pipeline for this task. One of the pipelines is BERTimbau [Souza et al. 2020], the Portuguese version of BERT that uses a neural architecture based on attention models and has been reported as the state-of-the-art for several text classification tasks [Devlin et al. 2019]. We fine-tune BERTimbau for our problem and compare it to conventional text classification pipelines. We also seek to define which aspects are domain-dependent and domain-independent when we apply the proposed method.

Therefore, the contributions of this paper are:

1. A new algorithm, `DEBACER`, which separates speech blocks from debates with a moderator;
2. A comparative study among conventional pipelines and BERTimbau for the `DEBACER` task, defining the best of them and evaluating domain dependencies;
3. The application of `DEBACER` in a set of minutes of the Portuguese Parliament.

2. Related Work

In this section, we present some relevant work published that uses machine learning and natural language processing techniques in debates with a moderator.

Guerini, Strapparava, and Stock [Guerini et al. 2008] propose the tagging of political speeches with audience reactions for further automatic analysis. The reaction acts as a validation of the rhetoric of a political party. The authors search for a set of keywords identifying the audience's state and then apply the TextPro and SentiWordNet to compute the persuasive impact of the speech. In our work, we want to identify automatically the speech transition among several politicians. In the minutes, there are also some reactions that we will use for further research on persuasive speech.

Yu, Kaufmann, and Diermeier [Yu et al. 2008] developed a framework for classifying party affiliation from political speeches. The authors trained Naive Bayes and SVM classifiers using the 2005House dataset to validate the Senate speeches. The authors found that the speeches contain a time-dependency pattern and more recent data drives thru a better classification. Our work differs on identifying individuals, and not political parties. In terms of techniques, we also employ a more sophisticated classification pipeline using BERTimbau.

Lippi and Torroni [Lippi and Torroni 2016] presents an automatic extraction algorithm to capture arguments and claims from UK politicians. The pipeline is composed of three modules speech recognition system, feature extraction (Bag-of-Words, part-of-speech tags, and lemmas), and a classifier (SVM). Our proposed pipeline is similar to this work, but we aim to identify the speaker in a specific part of the speech. We use more robust classifiers that can compete with state-of-the-art models (BERTimbau).

Roush and Balaji [Roush and Balaji 2020] proposes a model called debate2vec, which used a trained model using a dataset containing text from public debates on the parliament. The model uses a set of fast text word vectors previously described by Bojanowski *et al.* [Bojanowski et al. 2017]. The focus of the model is to classify arguments on political speeches correctly. Our work differs in identifying individuals rather than political arguments.

3. Proposal

Speeches given during a meeting follow a particular chronology and are transcript into minutes. Each *speech* $\mathcal{S}$ has a *debater* $\mathcal{D}$ who has the floor, and its content is composed of a sequence of n uttered *words* $\mathcal{S}_\mathcal{D} = (w_1, w_2, w_3, \ldots, w_n)$. We denote an *agenda item* $\mathcal{A}$, as a sequence of m speeches $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \ldots, \mathcal{S}_m)$ related to the same meeting item. A *minute* $\mathcal{M}$ is the sequence of all agenda items, $\mathcal{M} = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_l)$ that make up a given meeting, having a unique identification for each meeting.

It is usual for a person's speech to either be a statement about something discussed immediately preceding speech or introducing a new subject. Our objective is to partition the speeches in the political statements to identify a sequence of speeches that refer to the same subject. To do so, we define a *speech block* $\mathcal{B}_{ij} = (\mathcal{S}_i, \mathcal{S}_{i+1}, \ldots, \mathcal{S}_j)$ as a subsequence of $\mathcal{A}$, such that $\forall k \in (i, j]$, $\mathcal{S}_k$ is a speech that follows $\mathcal{S}_i$ logically, i.e., a block refers to speeches about the same subject. The purpose of the algorithm proposed here, DEBACER, is precisely to partition each agenda item $\mathcal{A}_x$ into a sequence of blocks $\mathcal{B}^x_{ij}$, such that $\mathcal{B}^x_{1i} \cap \mathcal{B}^x_{ij} \ldots \cap \mathcal{B}^x_{mn} = \emptyset$ and $\mathcal{B}^x_{1i} \cup \mathcal{B}^x_{ij} \ldots \cup \mathcal{B}^x_{mn} = \mathcal{A}_x$.

3.1. The DEBACER Algorithm

From a set of minutes $\Pi_\mathcal{M}$ containing transcripts of moderated debates, DEBACER divides all agenda items into blocks of speech. For each agenda item, $\mathcal{A}$ inside each minute $\mathcal{M}$, DEBACER runs through all moderator's speeches and identifies if their content indicates an interruption in the subject. Two functions are used (step 8 in Algorithm 1):

ISMODERATOR: Check if the debater is the moderator. It returns TRUE when the debater of $\mathcal{S}$ is the current moderator (*i.e.*, the current chairperson). This information is extracted from the database, where there is a special marker to indicate who the moderator is.

IsSubjectInterruption: Check if the content is classified as an interruption in the subject. It uses a domain-dependent text classification pipeline $\mathcal{C}$ to find out whether the moderator's speech content matches an interrupt or not.

A new speech block starts if the moderator changes the subject (steps 9 to 12). Otherwise, the speech belongs to the current speech block (step 13). At the end of the process, the database contains labels from the speech block each speech of the political declarations belongs, of each of the minutes in $\Pi_{\mathcal{M}}$ (step 17).

Algorithm 1 Split the political statements from the minutes into speech blocks

Inputs: a set of minutes $\Pi_{\mathcal{M}}$ (the database), and a subject interruption classifier $\mathcal{C}$

```
1: procedure DEBACER(Π_M, C)
2:     for each M ∈ Π_M do
3:         for each A ∈ M do
4:             A.blocks ← {}
5:             i ← 0
6:             B_i ← {}
7:             for each s ∈ A do
8:                 if IsModerator(s.debater) ∧ IsSubjectInterruption(C, s.content) then
9:                     A.blocks ← A.blocks ∪ B_i
10:                    i ← i + 1
11:                    B_i ← {}
12:                end if
13:                B_i ← B_i ∪ {s}
14:            end for
15:            A.blocks ← A.blocks ∪ B_i
16:        end for
17:        Update(Π_M, A)                ▷ Updates the database Π_M with A sliced
18:    end for
19: end procedure
```

3.2. Domain-dependent Aspects

Although DEBACER is domain-independent and can be applied to problems that match the proposed moderated debate problem description, it requires a properly arranged database and a domain-dependent text classification pipeline to detect interruptions. One premise is that the database in which the DEBACER will be executed is composed of 1 column with textual data (the speeches $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \ldots, \mathcal{S}_m\}$ delivered) and 1 column with its corresponding authors (the *debaters* $\mathcal{D}_i$). Considering a database in this format, it is essential to train the specific classification pipeline for the data domain. We can achieve that by annotating a training dataset from the database and applying it to a pipeline of supervised learning methods. These methods, at the end of the training, should classify "1" for the interruption – when a speech block $\mathcal{B}_{ij}$ ends and another block $\mathcal{B}_{(j+1)k}$ is initiated – and "0" otherwise. In the experiments in Section 4, we evaluate several pipelines and recommend the best one for this, which may involve text cleaning, feature selection, and different classifiers.

A fundamental aspect to be considered when training the classifier is that, in general, the data will be *inherently imbalanced*, that is, an uneven distribution of target groups is a characteristic of the problem. In the case of a debate, in speaker transitions, there is more continuation ("0") than subject change ("1"). Not properly dealing

with this problem can lead to classifier bias and poor performance. It is possible to treat the imbalance problem at two levels: data and algorithm. Data-level methods can take into account techniques such as *Stratified K-fold Cross-validation* (CV) and a frequency matching of classes applying *Dataset Resampling* or *Data Augmentation*. On the other hand, algorithmic-level methods will take into account the balancing mechanisms of the training algorithms for each type of model (for example, Random Forest applies balanced sub-sampling) and which performance metrics are used to compare them, especially while doing hyperparameters search. In section 4, we apply some of these methods: a modified version of K-fold CV, algorithms that somehow deal with data imbalance (*BERT, LR, SVM, RF*), and smart metrics for this problem: *F1-Score*, *Cross-Entropy* and *Brier Score*. This new pipeline should then be applied to the database to determine which speeches delivered by the moderator are interruptions and which are not.

4. Experimental Setup

Our experiments aim to validate the algorithm proposed in Section 3.1, as well as to find the best pipeline for the moderator's speech classifier. To this end, we make use of minutes of sessions of the Portuguese Parliament and compare BERTimbau, which has presented the best results in the literature for classification of texts in Portuguese, with four other common features for language processing.

4.1. Data

Data Collection We leverage data from the Portuguese Parliament (*Assembleia da República*) website[1] by using a web crawler algorithm specifically designed for the task of downloading the minutes in TXT format. These minutes were then separated into individual speeches, which were organized into a structured database, consisting of the fields: minute id, date, speaking order, debater, party, text, and agenda item, as illustrated in Figure 1. The current legislature minutes were used in this work, the XIV Legislature of the Portuguese Republic (from 2020/09/16 to 2021/02/25). Once ready, the database was composed of 20543 rows. For DEBACER application purposes, we selected from this database only the agenda item "political statements" that correspond to the moment of the parliamentary meeting in which members can openly discuss different topics, introducing new subjects or commenting on previous ones.

Minute ID	Date	Speaking Order	Debater	Party	Text	Agenda Item
DAR-001	17 DE SETEMBRO DE 2020	216	Presidente	Fernando Negrão	Retomando as declarações políticas, dou agora ...	declarações políticas
DAR-001	17 DE SETEMBRO DE 2020	217	João Gonçalves Pereira	CDS-PP	Sr. Presidente, Sr.as e Srs. Deputados: No fin...	declarações políticas

Figure 1. Transcripts Database.

Data Annotation It is necessary to annotate a dataset in order to apply it to a supervised learning pipeline that, at the end of the training, classifies "1" for the interruption and "0" otherwise. In the case of the Portuguese Parliament, the annotation process was carried

[1] https://debates.parlamento.pt/

Table 1. Target variable distribution. 1 is for subject interrupt and 0 otherwise.

Target Variable	Moderator					Total
	José M. Pureza	Eduardo F. Rodrigues	Edite Estrela	António Filipe	Fernando Negrão	
#0	165	147	99	69	69	**549**
#1	10	14	5	7	5	**41**
Total	**175**	**161**	**104**	**76**	**74**	**590**

out in a semi-automatic mode. We sampled and manually labeled about 70 speeches pronounced by chairpeople (*i.e.*, the moderator). This annotated data was used to train a Random Forest classifier which was used to label the remaining unlabeled data. Then, a manual review was carried out, observing each chairperson's speech and its context (speeches of previous and subsequent debaters) to correct misclassifications and revalidate the premise that we could reduce the partitioning speeches only based on the chairperson's speeches. The labeling process with the Random Forest classifier considerably reduced the human efforts involved in the annotation. Table 1 presents the distributions of the resulting annotated dataset.

Multi-label Stratified K-fold Cross-validation is a widely used technique to assess the generalization of a model. In particular, the K-fold is a well-known cross-validation method that consists of randomly dividing the dataset into K non-intersecting (mutually exclusive) sets, then training the model K times, covering all possible combinations of having the union of $K-1$ folds as the training set and the remaining fold as the test set. However, some problems present us with situations in which the uniformity of these K folds may be impaired: it may happen because the problem has more than one target variable or, in the Portuguese Parliament case, because factor others than the target variable influence the data distribution. We specifically mention two factors that may affect the performance of the classifier: (i) the debater, because each person has, within its individuality, its vocabulary preference; and (ii) the time, language is dynamic, and overtime terms become outdated, and new ones appear throughout years of parliamentary sessions. [Sechidis et al. 2011] proposes a stratified K-fold for multiple variables to have a population distribution in the subgroups more faithful to the parent group. Considering the short time interval between the minutes processed, we chose not to consider the time variable in this problem. Instead, **we employ the Multi-label Stratified K-fold approach, only taking the debater variable ($\mathcal{D}^*$) and the target variable (0 or 1) as labels**.

4.2. Baselines

BERTimbau: The BERT architecture consists of 12 Transformers blocks, each block has a hidden size of 768 and 12 self-attention heads. We fine-tuned BERTimbau pre-trained model, adding three dense layers (64 ReLU-32 ReLU - 1 Sigmoid), with a dropout of 0.2 between them, with all BERT layers unfrozen (totalizing about 334M training parameters), learning rate of 10^{-5}, AdamW optimizer, and Binary Cross-Entropy as the loss function.

Bag-of-Words (BoW): the most straightforward text feature. It is a sparse vector of the frequency of words in a text, whose size is equal to the vocabulary size.

Bag-of-N-Grams (BoNG): a derivation of BoW, a vector of frequencies of the N-Grams present in the text, *i.e.*, the count of all possible appearances of specific N words in a row. We use $N = 3$, *i.e.*, unigrams, bigrams, and trigrams were used in our frequency vector. The size of this vector can be large, so we use feature selectors to choose which N-Grams are most relevant. Some classification algorithms already have this built-in (like Random Forest), but when it is not, we employ *Truncated SVD* [Kim et al. 2005], which applies Single Value Decomposition for dimensionality reduction in sparse matrices. So the BoNG acts as a frequency vector of relevant expressions.

Word2Vec: in this configuration, we train a *Word Embedding* representation of each word on the entire base of Portuguese Parliament minutes (from 2020/09/16 to 2021/02/25), using *word2vec* method [Mikolov et al. 2013]. We use size $n = 100$ and take the average of the vectors to provide a *Sentence Embedding* representation of each data.

Doc2Vec: we train a *Sentence Embedding* representation of each data, also on the entire base, using the *doc2vec* method [Le and Mikolov 2014] with $n = 50$.

For **BoW**, **BoNG**, **Word2Vec** and **Doc2Vec** configurations, the data is pre-processed before being used for training. We employ *Tokenization* (segmenting a text into small significant units), *Stopword Removal* (cutting non-significant parts of the vocabulary such as articles, connectives, prepositions) and *Lemmatization* (converting nouns and adjectives to their masculine and singular form and transforming the existing verbs into their infinitive form in order to reduce vocabulary size and promote the abstraction of the word meaning). For these configurations, at the end of the pipelines, we apply different classifiers that have been reported to perform well for imbalanced data text classification: **Logistic Regression (LR)** [Fernández et al. 2018], **SVM** [Liu et al. 2009] and **Random Forest (Random F.)** [Wu et al. 2014]. Figure 2 outlines the pipeline configurations to be evaluated in this experiment.

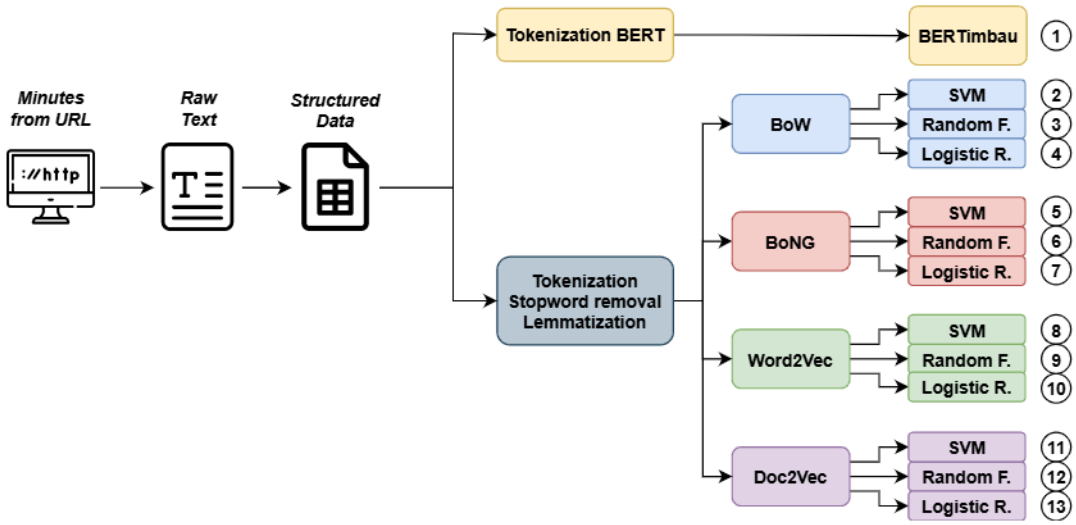

Figure 2. Scheme representing the pipelines configurations to validate DEBACER.

4.3. Performance Metrics

Considering the inherently imbalanced nature of the data in this task, we chose metrics that help offset this problem and compare models more fairly. The following three metrics

are used, in order of importance:

F1-score: evaluating based on accuracy tends to disregard the minority class. This can be solved using $Recall = \frac{TP}{TP+FN}$. However, using only recall makes us lose control of the prediction quality, so we define $Precision = \frac{TP}{TP+FP}$. The F1-score is the harmonic mean between both. So, when we maximize the F1-score, we maximize the gains in both properties.

Cross-entropy: quantifies the average difference between predicted and expected probability distributions, being defined as $CE = -\frac{1}{N}\Sigma_i(1-y_i)log(1-f(x_i)) + y_i log(f(x_i))$, where y_i is the target probability and $f(x_i)$ is the learned function.

Brier Score: measures the certainty the classifier has in its predictions, computing the mean squared error (MSE) between predicted probability scores and the true class indicator, where the positive class is coded as "1", and negative class as "0" [Fernández et al. 2018]. We use a modified version of this metric, proposed by [Wallace and Dahabreh 2012], that consists of decomposing this metric for each class, which for the positive would be defined as $BS^+ = \frac{1}{N_{y_i=1}}\Sigma_{y_i=1}(y_i - f(x_i))^2$.

Wilcoxon-Holm post-hoc analysis [Ismail Fawaz et al. 2019] is used to compare the performance of the pipelines used against BERTimbau. The test significance is $\alpha = 0.05$.

4.4. Implementation Details

The `DEBACER` algorithm and the Portuguese parliament minutes database were implemented using the Pandas library. We implemented BERTimbau from Hugging Face using TensorFlow framework. Other baselines made use of Scikit-learn library. We performed hyperparameter optimization by Bayesian search using Tune framework[2]. **We use the F1-score metric for model ranking and selection**. Winning models for each configuration are shown in Table 2. The experiments were run on an Intel Xeon 2-core 2.30 GHz CPU and a Nvidia T4 16 GB 1.59 GHz GPU. Table 2 shows the winner hyperparameters for each pipeline configuration.

5. Results and Discussion

Table 3 shows the results of the pipelines compared for detecting subject change. When comparing the proposed baselines, it is remarkable that BERTimbau immediately presents an excellent performance: it shows a good F1-Score (97.5 %), the lowest Cross-Entropy (0.010) and one of the lowest Brier Score Positive (0.025). Smaller Cross-Entropy means that the model generalized better the problem as a whole. The lower the Brier Score Positive, the more confident the classifier is in its predictions about the target class ("1"). However, the best comparison metric is the F1-Score, and in this configuration, the sparse representation by Bag-of-N-Grams performed numerically above BERTimbau (with 97.8 %), while the sparse representation by Bag-of-Words tied with BERTimbau. The continuous representations (doc2vec and word2vec) performed numerically slightly below the others. Nevertheless, Wilcoxon-Holm post-hoc analysis by pairwise statistical difference comparison presents a statistical tie between the best versions achieved for each pipeline. The critical difference diagram is in Figure 3.

As for conventional pipelines, it is worth mentioning that the Random Forest classifier had a much lower performance for continuous textual representations (doc2vec and

[2]https://docs.ray.io/en/master/tune/

Table 2. The winning hyperparameters for each pipeline configuration

Features	Best Classifier	Hyperparameters
BoW	SVM	C=482.2 kernel='linear'
	Random F.	criterion='gini' class_weight='balanced_subsample' n_estimators=357
	LR	solver='LBFGS' penalty='L2' class_weight=None C=1.02
BoNG	SVM	n_TSVD=188 kernel='linear' C=7932
	Random F.	criterion='gini' class_weight=None n_estimators=680
	LR	n_TSVD=148 solver='SAGA' penalty='L1' class_weight=None C=20.1
word2vec	SVM	kernel='rbf' C=3.75
	Random F.	criterion='entropy' class_weight=None n_estimators=238
	LR	solver='saga' penalty='L1' class_weight=None C=37.05
doc2vec	SVM	kernel='linear' C=176.36
	Random F.	criterion='gini' class_weight=None n_estimators=231
	LR	solver='LBFGS' penalty='L2' class_weight='balanced' C=97448

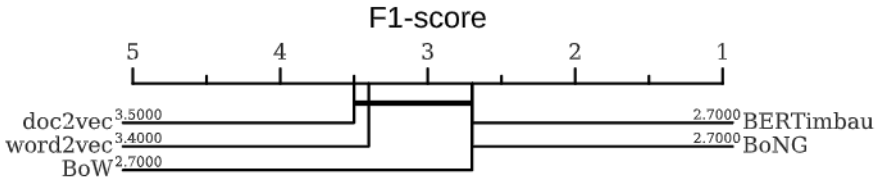

Figure 3. Critical difference diagram showing pairwise statistical difference comparison of the five pipeline configurations for detecting subject change.

word2vec), which was expected due to the tree learning mechanism. It is noticed that its performance on sparse representations were very similar to the other algorithms, as they would naturally be easy to be divided by a composition of axis-parallel decision boundaries. SVM and LR had very similar performances in both continuous and sparse representations, but if we take F1-score as the first criterion, and Cross-Entropy and Brier Score as a tiebreaker, Logistic Regression wins in all conventional pipelines, but word2vec.

The excellent performance of pipelines that use sparse features is justified by a critical detail of the nature of the problem: there are words (BoW) and expressions (BoNG) that are a strong indication of whether the current subject is being interrupted or not. **These act as triggers, leading the classifier directly to the decision**. This is why weights, such as the commonly used TF-IDF, are not justified for this problem. It should be noted that **all models** (BERTimbau, SVM, LR and Random Forest) **were able to deal with the class imbalance problem**. At the data level, Stratified K-fold CV was applied, and at the algorithm level, the F1-score metric was used to rank the models, in addition to the mechanisms that the models have for this: the complexity of BERT (with 334M parameters), balanced sub-sample from Random Forest and balanced class weighting from LR.

Table 3. Experimental results

Features	Classifier	F1-score	Cross-Entropy	Brier-Score +	Time
BERT Tokens	BERTimbau	0.975 ± 0.031	0.010 ± 0.007	0.025 ± 0.034	$4min40s$
BoW	SVM	0.968 ± 0.063	0.019 ± 0.016	0.025 ± 0.020	$0.1s$
	Random F.	0.975 ± 0.031	0.036 ± 0.007	0.067 ± 0.027	$3.1s$
	LR	0.975 ± 0.031	0.025 ± 0.012	0.038 ± 0.016	$0.1s$
BoNG	SVM	0.978 ± 0.044	0.020 ± 0.010	0.041 ± 0.020	$1.4s$
	Random F.	0.976 ± 0.029	0.041 ± 0.005	0.103 ± 0.011	$5.1s$
	LR	$\mathbf{0.978 \pm 0.044}$	$\mathbf{0.018 \pm 0.009}$	$\mathbf{0.037 \pm 0.021}$	**3.8 s**
word2vec	SVM	0.936 ± 0.058	0.028 ± 0.010	0.067 ± 0.074	$0.2s$
	Random F.	0.889 ± 0.055	0.067 ± 0.002	0.183 ± 0.053	$10.4s$
	LR	0.924 ± 0.048	0.351 ± 0.219	0.010 ± 0.006	$5.3s$
doc2vec	SVM	0.936 ± 0.060	0.036 ± 0.020	0.104 ± 0.095	$0.1s$
	Random F.	0.678 ± 0.126	0.099 ± 0.010	0.279 ± 0.062	$2.5s$
	LR	0.948 ± 0.053	0.234 ± 0.219	0.007 ± 0.006	$0.3s$

Another critical issue is that **BERTimbau handled the European Portuguese dialect well**. It is known that the model is pre-trained in the database brWaC [Souza et al. 2020], which is mainly composed of data from the Brazilian Portuguese dialect. However, as the language is the same and the fine-tuning was performed with the transformer layers unfrozen, it was possible to adapt the pre-training for the specific domain (legislative debates) and the european dialect. On the other hand, it is remarkable that the BERT fine-tuning task can be costly and unnecessary, depending on the problem. Table 3 shows that the BERTimbau fine-tuning time is much higher than the training time of the other models. BERTimbau also makes greater use of memory space during fine-tuning and runtime. The execution time is known to be longer as well. Thus, **it is possible to say that we achieved the same results as the state-of-the-art in text classification with less computational effort**. Therefore, it is also recommended to consider traditional pipelines for tasks of the same type.

Finally, the results indicate that the **best classification pipeline for `DEBACER`** in the case of the Portuguese Parliament is the one that uses **Bag-of-N-Grams as text representation and Logistic Regression as classifier**. This is based not only on the performance metrics it obtained, but mainly on being **the best trade-off between performance and computational resources spent** (time and memory in training and execution). Furthermore, in any pipeline configuration the performance metrics results were good, which **demonstrates the viability of `DEBACER` as a solution to the proposed problem of partitioning moderated debates**.

6. Conclusion and Future Work

In this paper we proposed `DEBACER`, a debate slicer method for partitioning speeches into blocks that share a common stage of a discussion. `DEBACER` groups speeches from different debaters that refer to the same subject. This method can be applied to moderated dialogues, where a moderator controls the session and passes the floor to whoever is the next to speak. The output blocks of our method allow the execution of other NLP tasks, such as topic modeling and sentiment analysis, as well as an assertive quantification of statistical information related to these data, such as citations to a specific topic or subject

of interest, and its context.

There are many domains of DEBACER-able data. Among them, we can cite trials, public hearings, parliamentary sessions, and electoral debates, all sharing the structure of a moderated dialogue. For working properly in these domains, our algorithm's classification pipeline needs training within data from that domain, after which it will be ready to classify the moderator's speeches, and partitioning the data into blocks. In this work, DEBACER was validated on data from the minutes of the Portuguese Parliament. We evaluated different pipelines to assess if the BERTimbau architecture, state-of-the-art in this task, is significantly better than more traditional ones. We show that a classic pipeline achieves scores statistically similar to BERTimbau, but with the advantage of having faster execution and training times, and less memory usage.

Some directions for future work will involve evaluating the performance of DEBACER and the techniques presented in this work for new datasets from different domains in order to validate their strength. In addition, we intend to explore applications where the proposed algorithm can be helpful as an intermediate step for NLP tasks in these datasets, such as topic modeling and opinion mining. Moreover, we want to investigate the potential and viability of employing cross-domain generalization strategies towards a universal classifier for DEBACER.

Acknowledgments

This research was supported in part by *Itaú Unibanco S.A.*, with the scholarship program of *Programa de Bolsas Itaú* (PBI), and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, and CNPQ (grant 310085/2020-9), Brazil. Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policy or position of the Itaú-Unibanco, CAPES and CNPq.

References

Ales, Z., Pauchet, A., and Knippel, A. (2018). Extraction and clustering of two-dimensional dialogue patterns. *International Journal on Artificial Intelligence Tools*, 27(02):1850001.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the ACL*, 5:135–146.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from imbalanced data sets*, volume 11. Springer.

Guerini, M., Strapparava, C., and Stock, O. (2008). Corps: A corpus of tagged political speeches for persuasive communication processing. *Journal of Information Technology & Politics*, 5(1):19–32.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.

Kim, H., Howland, P., Park, H., and Christianini, N. (2005). Dimension reduction in text classification with support vector machines. *Journal of machine learning research*, 6(1).

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Lippi, M. and Torroni, P. (2016). Argument mining from speech: Detecting claims in political debates. In *30th AAAI Conference on Artificial Intelligence*.

Liu, Y., Loh, H. T., and Sun, A. (2009). Imbalanced text classification: A term weighting approach. *Expert systems with Applications*, 36(1):690–701.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceeding of the ICLR*, pages 1301–3781.

Roush, A. and Balaji, A. (2020). Debatesum: A large-scale argument mining and summarization dataset. In *7th Workshop on Argument Mining*, pages 1–7.

Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer.

Shan, Y., Li, Z., Zhang, J., Meng, F., Feng, Y., Niu, C., and Zhou, J. (2020). A contextual hierarchical attention network with adaptive objective for dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6322–6333.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 403–417. Springer.

Wallace, B. C. and Dahabreh, I. J. (2012). Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 IEEE 12th International Conference on Data Mining*, pages 695–704.

Wen, T., Vandyke, D., Mrkšíc, N., Gašíc, M., Rojas-Barahona, L., Su, P., Ultes, S., and Young, S. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 1, pages 438–449.

Wu, Q., Ye, Y., Zhang, H., Ng, M. K., and Ho, S.-S. (2014). Forestexter: an efficient random forest algorithm for imbalanced text categorization. *Knowledge-Based Systems*, 67:105–116.

Yu, B., Kaufmann, S., and Diermeier, D. (2008). Classifying party affiliation from political speech. *Journal of Information Technology & Politics*, 5(1):33–48.