Mitigating Topic Bias when Detecting Decisions in Dialogue

Vanja Mladen Karan, Prashant Khare, Patrick Healey, and Matthew Purver
Cognitive Science Research Group, Queen Mary University of London, London, UK
{m.karan, p.khare, p.healey, m.purver}@qmul.ac.uk

Abstract

This work revisits the task of detecting decision-related utterances in multi-party dialogue. We explore performance of a traditional approach and a deep learning-based approach based on transformer language models, with the latter providing modest improvements. We then analyze topic bias in the models using topic information obtained by manual annotation. Our finding is that when detecting some types of decisions in our data, models rely more on topic specific words that decisions are *about* rather than on words that more generally indicate decision making. We further explore this by removing topic information from the train data. We show that this resolves the bias issues to an extent and, surprisingly, sometimes even boosts performance.

1 Intro

We spend a lot of our time in meetings. Recordings of such meetings in the form of video or audio recordings or transcripts can be a valuable resource, but we need automatic processing and summarization methods if we are to be able to quickly search and retrieve the information we need. According to user surveys, the primary requirement of users from a meeting summarization system is a record of the decisions made (Lisowska et al., 2004; Banerjee et al., 2005). It can allow tracking of decisions and the reasoning behind them, as well as alternatives that were proposed and discussed.

Previous work on the task of automatic decision detection (e.g. Hsueh and Moore, 2007; Fernández et al., 2008; Bui and Peters, 2010) shows that the problem is challenging: performance is limited (Fernández et al., 2008) unless strong assumptions about the nature of the data are made (Bui and Peters, 2010). E.g., assuming particular structure of the dialogue, rather than learning it from data. One reason for this is the lack of large datasets for the task. Here, we show that previous models are also affected by another issue resulting from lack of data: *topic bias*. The intuition behind this problem is that the models might pick up on words that decisions are *about* instead of words that generally indicate decision making. As an example of this we provide the decision utterance - *We agree to use a battery as a power source*. A decision detection model might pick up on *battery* or *power source* as indicating decision making, simply because these phrases are something that often accompany decision in our data. However, a more unbiased model would ideally pick up on *we agree* as indicating decision making. Our goal here is to explore and mitigate this problem by manually removing topic specific words, preventing the model from becoming topically biased.

The contributions of this paper are two-fold. First, we present a deep learning based prediction model for a decision detection task. Second, we give an analysis of topic bias in the data and models for this task, and show how our model can be made less susceptible to this bias compared to previous approaches. We make all our code and data publicly available.[1]

2 Related Work

Some work in decision detection treats it as a text classification problem, and in some domains this is successful; Bhat et al. (2017) achieve good accuracy detecting software architecture decisions in issue tracking systems. The same approach can be applied to face-to-face meeting dialogue, classifying individual utterances as decision-related or not on the basis of a range of lexical, structural and semantic features; but in this domain performance is lower (Hsueh and Moore, 2007). Fernández et al. (2008) improve on this by considering the structure of the decision-making dialogue: they propose a set of decision-specific dialogue acts (DDAs) and

[1] https://github.com/mladenk42/decibert

Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 542–547
July 29–31, 2021.

a model using support vector machines (SVMs) to classify each DDA, using the outputs to predict decision discussion regions. Similarly, Frampton et al. (2009) explore real-time decision detection.

Further improvements have been shown via more explicit modeling of decision-making dialogue structure, encoded as probabilistic graphical models, and including non-lexical and prosodic features (Bui et al., 2009; Bui and Peters, 2010), but at the cost of assuming a fixed structure to a discussion rather than learning it from data.

In contrast to related work, our primary focus is exploring the, thus far unaddressed, topic bias issues rather than maximum performance. Consequently, we opt for simpler models that use only the text without additional features. We include one traditional and one deep learning based model.

3 Dataset

We use the dataset introduced by Fernández et al. (2008), an annotated subset of transcripts from the AMI meeting corpus (McCowan et al., 2005) covering 17 meetings in which actors stage a simulated meeting with the task of *designing a remote control*. Each utterance is annotated with one or more of four specific *decision dialogue acts (DDAs)*: *issue* (I), *resolution proposed* (RP), *resolution restated* (RR), and *agreement* (A). Categories RR and RP are both very low in number, which would likely hinder deep learning approaches. However, they are conceptually very similar, so we decided to merge them into a single category we denote as R. The annotations are multilabel (one utterance can perform more than one DDA), although it is quite rare for an instance to have multiple labels (less than 1%). Other available utterance metadata includes speaker id, timestamp, and a decision id (only for DDA utterances). The total number of utterances in the dataset is 15680. DDAs are rare, with each category making up only 1-2% of utterances. The sparsity of the decision acts presents a considerable problem for all work on this data set. Table 1 gives some examples and statistics.

	count	%	example
I	138	0.9	And what tha what about the uh materials?
R	209	1.3	So I guess the case would be plastic,
A	324	2.1	Yeah. Uh as an option maybe.

Table 1: Utterance counts and percentages for the three DDA categories – Issue (I), Resolution (R), and Agreement (A), with examples.

4 Methodology

As part of our methodology, we next describe the models and evaluative metrics we employ.

4.1 Models

Baseline As features for the baseline model, we generate a Tf-Idf weighted vector representation of each utterance. Then, we use a similar baseline as the one in (Fernández et al., 2008). We include context by extending the vector of each utterance with vectors of nearby utterances in a context window of size N around it. We feed the extended representations into a logistic regression classifier.

BERT-LSTM As the basis of our deep learning approach we use BERT, a popular transformer-based language model shown to perform well across a diverse range of tasks (Devlin et al., 2019). Specifically we use SentenceBERT (Reimers and Gurevych, 2019) to generate a 768-dimensional vector representation for each utterance. To generate a prediction for utterance u_k at position k, given a context window of size N, we consider the sequence of BERT vector representations for utterances $u_{k-\frac{N}{2}}...u_{k+\frac{N}{2}}$, of length N. We run a bidirectional long-short term memory (LSTM, Hochreiter and Schmidhuber, 1997) network over this sequence, yielding N hidden state outputs.[2] Each output is fed into 3 separate linear + softmax layers, producing three separate binary decisions, one for each DDA class.[3] Thus, for each utterance we obtain, as a byproduct, a multilabel decision for each utterance within its context window.

When training the model we minimize the following loss function:

$$\sum_{c\in\{I,R,A\}} \sum_{k=1}^{K} \sum_{j=k-\frac{N}{2}}^{k+\frac{N}{2}} CE_w(y_{c,k,j}, t_{c,k,j}) \qquad (1)$$

where c is one of the categories, k iterates over utterances, and j over context utterances of utterance u_k. Moreover $y_{c,k,j}$, denotes the prediction of the model for utterance u_j when it is part of a context window centered over u_k. This prediction can indicate u_j belongs to category c ($y_{c,k,j} = 1$) or does not ($y_{c,k,j} = 0$). The corresponding correct prediction is denoted as $t_{c,k,j}$. Finally, CE_w

[2] We could not consider each meeting as one long sequence, as there are only 17 of them.

[3] The linear layers share weights across all timesteps.

denotes the cross-entropy loss, weighted to account for the highly imbalanced number of positive and negative examples in each category.[4] We use this as it works with multilabel annotations.

When making predictions with this model for utterance u_k with respect to class c, we run the above model for a context window of size N around u_k and take the center prediction, i.e. $y_{c,k,k}$.

Since the goal of this paper is to explore bias, rather than maximize performance, we stick to this simpler deep learning approach and leave the investigation of more complex alternatives, such as dialog oriented models from (Wu et al., 2020; Gu et al., 2020) to future work.

Both models are implemented using Scikit-learn (Pedregosa et al., 2011) and PyTorch (Paszke et al., 2019). The hyperparameters and other training details of all models are provided in Section 5.

4.2 Evaluation metrics

The models are evaluated using the metrics of Fernández et al. (2008), with two evaluation setups described below.

Utterance level evaluation (ULE) This approach is implemented as described by Hsueh and Moore (2007). In essence it is a lenient variant of F-score that works on the level of individual utterances but tolerates a level of misalignment between the labeled DDAs and those hypothesized by the model: we use a window of ± 20 utterances around the gold utterance, following (Hsueh and Moore, 2007; Fernández et al., 2008).

Segment level evaluation (SEG) Here a meeting is split into fixed 30 second segments, with a segment considered as predicted positive if it contains at least one utterance labeled as positive for at least one DDA by the model. Gold labels for each segment are positive if (1) it overlaps with any gold annotated DDA or (2) the nearest gold annotated DDA before and after the segment have the same decision id. (Part (2) accounts for segments which are a part of decision discussion but do not themselves contain any DDAs). The score is then computed as a standard F1 score.

4.3 Masking topic words

As all meetings in the dataset are on the same topic of *designing a remote control*, we hypothesize there could be topic bias in the data or models. The AMI meetings cover a relatively small set of issues (e.g., power source, case material, button type, colour) and proposed resolutions (e.g., kinetic energy, rubber, background light, transparent). A classifier is therefore likely to learn to detect issues/resolutions via this domain-specific vocabulary rather than more generalisable patterns. To explore this hypothesis, we first fit a Naïve Bayes classifier to the data using binary word counts as features. We do this for each category separately, with the category being a binary target variable. We then observe the most probable words for the positive outcome. The results reveal a considerable number of such topic words present in the most influential 50 words. Some more statistics and examples are given in Table 2.

	#topic words	examples
I	14/50	controller, power, solar, graphical
R	6/50	batteries, option, system, internal
A	3/50	remote, control, lights

Table 2: Statistics of topic words in the 50 most probable words per class in a Naïve Bayes classifier.

To investigate the extent of this effect, we attempt to train less topic-dependent versions of our models. We first manually examined a total of 656 utterances labeled with at least one DDA category, resulting in a list of 115 domain-topic words.[5] We use this as a masking dictionary to produce two modified versions of the transcripts. First, with the masked words removed; second, with the masked words replaced by the special BERT [MASK] token. These are then used to train models which we hypothesize will show less topic bias. As the first method performs better, we present only results from the first due to reasons of space.

5 Experiments and results

Experiment setup We evaluate the models using leave one out cross-validation. In each iteration, we train the models on 16 meetings and test them on the remaining meeting. For both the ULE and SEG evaluation setups, scores are calculated at the level of the meeting and averaged.

For the logistic regression baseline, we optimize the regularization hyperparameter to maximize the

[4] We use the method of King and Zeng (2001) implemented in scikit-learn to obtain the weights.

[5] This was done completely manually, and is not related to the Naïve Bayes analysis, which we did only to gain intuition and motivation for the manual analysis.

	No masking						With masking					
	Baseline			BERT			Baseline			BERT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
I	.152	.440	.209	.221	.314	.232	.140	.530	.210	.237	.361	**.263**
R	.210	.713	.304	.236	.490	.292	.174	.769	.271	.294	.527	**.333**
A	.175	.845	.283	.257	.658	**.352**	.165	.844	.270	.255	.627	.343
SEG	.337	.885	.527	.419	.761	.540	.355	.906	.510	.427	.770	**.547**

Table 3: Results of the baseline and BERT models for all four classification setups with and without masking.

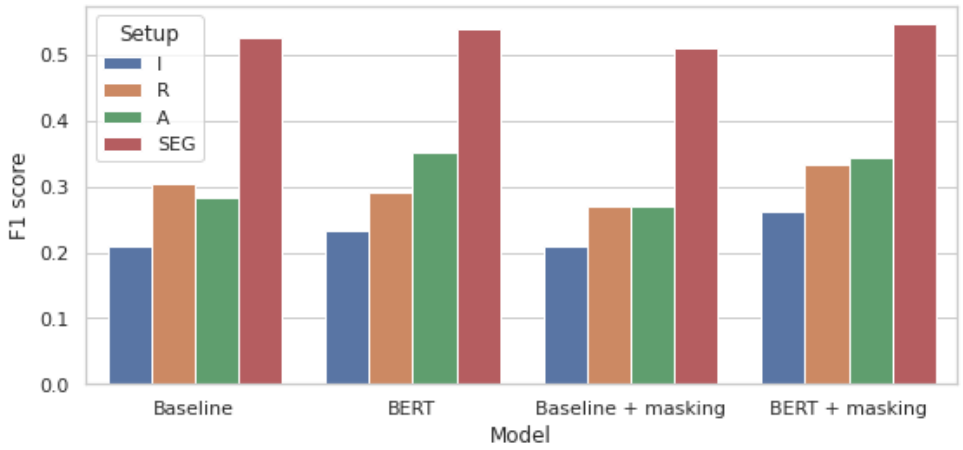

Figure 1: Model performances from Table 1 visualised.

overall crossvalidation score;[6] the best setting was 1.0. For the BERT-LSTM we optimize hyperparameters on held out data using a fixed split. We set the hidden layer size of the BiLSTM to 50 and the number of layers to 1. The best context window size was ± 1 for both models. We keep these settings fixed throughout the rest of the experiments.

For the BERT-LSTM model we use the Adam (Kingma and Ba, 2015) optimizer with learning rate 10^{-4} and minibatch size 32. Out of the 16 training meetings we set one aside as a development set for early stopping. We train the model until there has been no improvement in score for any of the evaluation setups on the development data for 5 consecutive epochs. Furthermore, we found that due to the small data set size, this training regime sometimes produces very bad models (depending on random initialisation). We circumvent this by training it several (in our case 16) times with different development meetings and different random initialisations. We use on the test set the variant that has highest test set confidence scores.[7]

[6]Making the baseline stronger than in a realistic scenario.
[7]This in no way uses the test set labels.

Results We give our main results in Table 3; note that the low absolute values are due to the rarity of DDA utterances. A visualisation of the same data is given in Figure 1. The BERT-LSTM model outperforms the baseline model in terms of F1 score for almost all cases, and consistently sacrifices recall to gain precision.

We next explore how masking affects each model. For the baseline, masking slightly reduces performance; although we know from Table 2 that many of the non-masked model's features will be topic-specific, the masked training seems to recover most of the performance.

For BERT-LSTM, however, performance increases: at least for some examples, removing topic bias from the data helps improve performance. Differences between non-masked and masked BERT-LSTM models are statistically significant ($p < 0.05$) for I, R, and SEG.

The improvements are largest for I and R categories, which use more topic-specific vocabulary; and are absent for the A category, which uses much fewer topic words. The SEG scores also modestly increase, as small improvements for individual utterances have some influence on the overall output.

To better understand this phenomenon in the

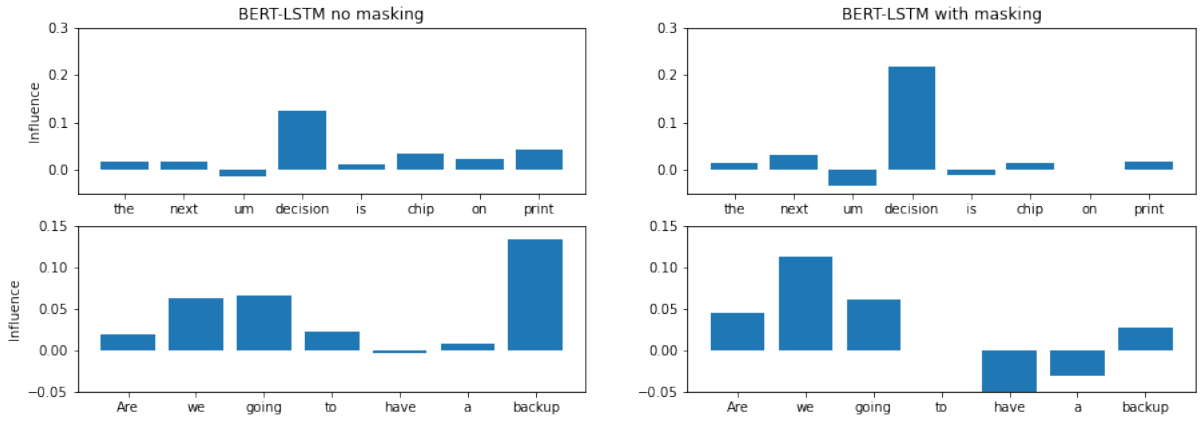

Figure 2: Feature influences derived by the LIME method for BERT-LSTM models without (left) and with (right) masking. Positive influence values denote a word is pushing the prediction towards the positive category, and vice versa for negative ones. Rows represent utterances.

BERT-LSTM model, we applied the LIME feature analysis method (Ribeiro et al., 2016). Figure 2 illustrates the results for two utterances.

For the first utterance, we see that after masking, the model relies much more on the word *decision* than on the domain-specific words *chip* or *print*. In this case masking corrected the output of the model from 0 to 1. In the second utterance, however, shifting the focus from the domain-specific *backup* to the more general *Are we going to* phrase, while seemingly desirable, causes a mistake changing the prediction from 1 to 0. We hypothesize this is due to lack of data to learn all decision indicative phrases properly. These insights and the results in Table 3 suggest that masking does, to an extent, mitigate the topic bias problems, but that small dataset size is still hindering performance.

6 Conclusion

We have explored the problem of topic bias in detecting decision dialogue acts (DDAs). In particular, we have identified bias for the Issue and Resolution types of DDAs. We experimented with mitigating the bias by manually identifying and removing topic related words and our main finding is that, while this partially mitigates the bias issues and sometimes even improves performance. However, to further confirm these findings more experiments on other, larger data data sets are required.

There are several avenues of future work. These include exploring models that capture speakers, using non-decision dialogue acts as additional information, or pretraining language models on decision-related sentences. The immediate direction, however, is to increase the size of DDA annotated data and include a more diverse set of topics.

Acknowledgments

This work was supported by the EPSRC under grant EP/S033564/1, Streamlining Social Decision Making for Improved Internet Standards. Purver is also supported by the European Union's Horizon 2020 programme under grant agreements 769661 (SAAM, Supporting Active Ageing through Multimodal coaching) and 825153 (EMBEDDIA, Cross-Lingual Embeddings for Less-Represented Languages in European News Media).

We thank the anonymous reviewers for their insightful comments. We especially thank Gareth Tyson, Ignacio Castro, and Colin Perkins for fruitful discussions and constructive feedback.

References

Satanjeev Banerjee, Carolyn Rosé, and Alex Rudnicky. 2005. The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In *Proceedings of the 10th International Conference on Human-Computer Interaction (CHI)*.

Manoj Bhat, Klym Shumaiev, Andreas Biesdorf, Uwe Hohenstein, and Florian Matthes. 2017. Automatic extraction of design decisions from issue management systems: a machine learning based approach. In *European Conference on Software Architecture*, pages 138–154. Springer.

Trung Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical mod-

els and semantic similarity. In *Proceedings of the SIGDIAL 2009 Conference*, pages 235–243.

Trung H. Bui and Stanley Peters. 2010. Decision detection using hierarchical graphical models. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 307–312, Uppsala, Sweden. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008. Modelling and detecting decisions in multi-party dialogue. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 156–163, Columbus, Ohio. Association for Computational Linguistics.

Matthew Frampton, Jia Huang, Trung Bui, and Stanley Peters. 2009. Real-time decision detection in multi-party dialogue. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1133–1141.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Pei-Yun Hsueh and Johanna D. Moore. 2007. What decisions have you made?: Automatic decision detection in meeting conversations. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 25–32, Rochester, New York. Association for Computational Linguistics.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political analysis*, 9(2):137–163.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Agnes Lisowska, Andrei Popescu-Belis, and Susan Armstrong. 2004. User query analysis for the specification and evaluation of a dialogue processing and retrieval system. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*.

Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, et al. 2005. The AMI meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88, page 100. Citeseer.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.