Enhancing the Automatic Classification of Metadiscourse in Low-Proficiency Learners' Spoken and Written English Texts Using SpanBERT

Anonymous ACL submission

Abstract

This study aims to enhance the automatic identification and classification of metadiscourse markers in English texts using SpanBERT. Metadiscourse is a commonly used rhetorical strategy in both written and spoken language to guide the addressees through the discourse. Due to its linguistic complexity and dependency on the context, automated metadiscourse classification is challenging. With a hypothesis that LLMs may handle complicated tasks more effectively than supervised machine learning approaches, we tune and evaluate SpanBERT on the task using a dataset totalling 602,386 tokens and annotated with 24 labels. The results show a clear improvement over the spaCy baseline of 0.77, with the model achieving an accuracy and F1 score of 0.86 and 0.90, respectively. However, overfitting due to the unbalanced category distribution is an important issue. Two less frequent categories record an F1 score of 0.00, highlighting the need for more balanced data representation.

1 Introduction

Metadiscourse (MD) is an essential rhetorical strategy in both speaking and writing that realizes two of the metafunctions of language proposed by Halliday (1994): the textual and interpersonal functions. MD that mainly has a textual function is used to form a cohesive and coherent text (Kopple, 1985). The textual dimension comprises transitions (*but*, *and*, *because*), frame markers (*firstly*, *in conclusion*, *the next point is ...*), code glosses (*in other words*, *namely*, *for example*), and so on. Textual MD markers often have fixed forms and consistent meanings, hence they pose relatively few challenges in automatic classification. Conversely, MD that is primarily interpersonal shows different features. Addressers use interpersonal MD to comment on the propositions and to involve the addressees in their discourse. Examples include but are not limited to hedges (*may*, *probably*, *I'm not sure ...*), boosters (*certainly*, *must*, *I believe*), and addressing the addressees (*You may end up thinking that ...*, *You may ask...*, *Can you hear me?*). This dimension is linguistically more complex as it involves multiple syntactic classes and has flexible span boundaries. The complexity undoubtedly leads to difficulty in automatic classification. Previous research using supervised methods reveals the performance gap between the two broad dimensions (dos Santos Correia, 2018; Alharbi, 2016). Classification of textual MD has yielded satisfactory accuracy but classification of interpersonal MD is lacking.

Automatic MD classification has barely been studied. The little research that has been done only explored supervised methods, and Large Language Models (LLMs) have not been used to the best of our knowledge. As LLMs encode a broader range of semantic, syntactic and contextual information due to the more complex architecture that is pretrained on various linguistic resources, this study aims to improve the state of the art in automatic MD classification with a transformer-based method.

The raw data used in this paper were sampled from the International Corpus Network of Asian Learners of English (ICNALE) (Ishikawa, 2023). They effectively represent the diversity of natural language by containing both spoken and written English, as well as data from native speakers and learners. A strong motivation behind the selection of the data is that, learners may use more MD markers compared to native speakers because they intentionally put more effort into linguistic and meta-linguistic matters (Ädel, 2006).

Given that MD markers are captured in spans of varied lengths, we choose SpanBERT (Joshi et al., 2019) as our model. It is the main LLM we are aware of with span-specific training objectives. SpanBERT has been shown to outperform BERT models in extractive question answering, coreference resolution, and relation extraction tasks by its

authors.

2 Background

This section extends the concept of MD to its classification. The classification first refers to a theoretical taxonomy, which can be used when labelling the raw data. Two challenges have been identified in the theoretical classification and are anticipated to lead to difficulties in automatic classification.

2.1 Metadiscourse taxonomies

We introduced two main broad categories of MD in the introduction. In linguistic practice, MD is classified into many categories but there is no uniform taxonomy. Along with the development of relevant research, scholars in the field came up with taxonomies from varied perspectives (Kopple, 1985; Crismore et al., 1993; Mauranen, 1993; Milne, 2003; Hyland, 2005; Ädel, 2006; Ädel, 2010). Among them, Hyland's (2005) taxonomy is the most commonly used one. It provides a list of discovered MD markers for English based on corpus data. However, we had a couple of concerns about adopting the taxonomy in this paper. Above all, it might not be sufficient because it is built on the observations of formal written language, including textbooks, students' writing, research articles, CEOs' letters, etc.. Thus, it may not include MD markers typical of spoken language, nor incorporate mistakes such as grammatical errors and misspellings often found in learners' casual communication.

Ädel's (2010) taxonomy is representative of MD in spoken language. More importantly, the identification of the categories in this taxonomy relies primarily on the functions of MD in the discourse. For example, how the topic is introduced, developed, and concluded. Nevertheless, this taxonomy also has a limitation that does not fully meet the research purposes of this study. Ädel's (2010) taxonomy requires high explicitness of MD markers. In other words, if a text span does not contain deictic words that refer to the discourse or interlocutors, it will not be counted as MD in this taxonomy but may still be MD in other taxonomies. Therefore, Ädel's (2010) taxonomy is selected as the basis of the annotation scheme in this paper but some categories from Hyland's (2005) taxonomy are added and adjusted. The final taxonomy we use is shown in Table 1. The dimensions of "Metalinguistic comments" and "Discourse organization" correspond to the textual MD, while the dimension "Writer-reader/speaker-listener interaction" aligns with the interpersonal MD.

2.2 Challenges in MD classification

Two features of MD pose challenges in MD classification by human annotators and predictably also in automatic classification. Firstly, MD is highly context-sensitive. A span can be MD in one context but not in another context. For instance, "so" functions as a MD marker when it indicates a causal relation between two clauses (Example 1.1). However, it is not an MD marker when it refers to a way something was described (Example 1.2). The ambiguity makes the identification of MD from propositional contents challenging.

Example 1.

- *1: All people in the restaurant would be affected by smoking **so** it should be banned.*
- *2: I don't think **so**.*

Secondly, a MD candidate may belong to more than one category. In Example 2, "I fully agree that" is a MD marker to show the speaker's attitude. Within it, there is another MD marker "fully" that is a booster.

Example 2.
I fully agree that smoking should be banned in restaurants.

Previous attempts to classify MD have used supervised methods such as Support Vector Machines (SVMs) (dos Santos Correia, 2018) and Decision Trees (dos Santos Correia, 2013). Automatic classification of textual MD has reached good levels in existing studies because of explicit syntactic and semantic features, while that of interpersonal MD has not yielded good results yet. The performance of LLMs on similar tasks has been underexplored. We hypothesize that SpanBERT outperforms traditional supervised machine learning methods.

2.3 SpanBERT

SpanBERT (Joshi et al., 2019) is the pre-trained (Wei et al., 2023) LLM that is used in this project, a version of the BERT (Bidirectional Encoder Representations from Transformers) language model. Similar to BERT, SpanBERT takes sequences of tokens as input. The difference between SpanBERT and BERT is in their training objectives. SpanBERT is trained with a span-boundary objective.

Dimension	Category	Label	Examples
Metalinguistic comments	Repairing	RPR	*I'm sorry...*
	Reformulating	RFL	*to put it differently...*
	Commenting	CMT	*... is a difficult question.*
	Clarifying	CLF	*I don't mean to say*
	Exemplifying	EXP	*for example*
Discourse organization	Managing topics	MNT	*I will focus on...*
	Organizing statements	ORS	*and; but; so*
	Providing Evidentials	PED	*according to*
	Enumerating	ENM	*first; at last*
	Endophoric marking	EDP	*As we can see in Chapter III, ...*
	Previewing	PRV	*We will discuss...*
	Reviewing	RVW	*As I said last time, ...*
Writer-reader /Speaker-listener interaction	Epistemic attitudes	EPA	*I agree that...*
	Hedges	HDG	*perhaps; might*
	Boosters	BST	*definitely; should*
	Speech act labels	SAL	*I argue that...*
	Managing comprehension	MNC	*You know what I mean.*
	Managing channel/audience discipline	MCD	*Can you hear me?*
	Anticipating the audience's response	AAR	*You may ask...*
	Managing the message	MNM	*What I want to emphasize is...*
	Imagining scenarios	IMS	*Suppose you are giving a speech...*

Table 1: List of MD categories, labels, and examples

This encourages the model to represent the relationships between tokens within a span. It predicts the entire span of tokens instead of individual tokens. This is useful for tasks that require representations of text chunks, such as named entity recognition, relation extraction, as well as our task of MD identification and classification. In BERT and SpanBERT, a token is not always equal to a single word. The WordPiece (Wu, 2016) tokenizer is used to split longer or less frequent words into subtokens, "play" and "##ing" from "playing" for example. The prefix in "##ing" indicates that the word is a suffix of another token. This process allows the models to efficiently handle a larger vocabulary and unfamiliar words, with a compact vocabulary. Words that are part of the vocabulary remain intact.

3 Methods and data

Our dataset comprises 2980 texts totaling 602,386 tokens, from the ICNALE corpus. The corpus is topic- and task-controlled. Three registers including monologues, dialogues, and written essays are sampled, considering diverse stylistic features.

The average and median number of words per text are 202.14 and 207 words respectively. The smallest text contains 29 tokens, while the longest

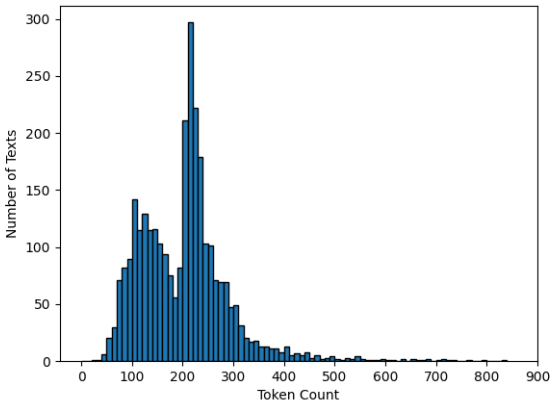

Figure 1: Distribution of token counts per text

text comprises 837 tokens. The big gap exists because the learner informants of the corpus have English proficiency ranging from A2 to B2 level according to the Common European Framework of Reference for Languages (CEFR). Learners with lower proficiency usually speak or write more slowly and thus produce shorter texts, and vice versa. Additionally, with an interest in improper usage of MD, three more labels are added to the annotation scheme:

- Misuse (MIS): The span is an MD marker but semantically it does not make sense in the current context.
- Error (ERR): The span is an MD marker but it contains grammatical errors or is misspelt.
- Uncertainty (UCT): The annotator is not sure if a span is an MD marker or not, or which category the span belongs to.

These linguistic issues may confuse LLMs if they happen frequently. The annotation scheme then consists of 21 MD labels and three other labels that are listed above. The highlights in the Figure 2 illustrate MD spans in the text.

EPA HDG BST
I don't agree with it. In my opinion, college students should be concentrated on their study.

Figure 2: Span example

The dataset has been manually annotated using Prodigy (Honnibal et al., 2024), an annotation tool for creating training and evaluation data for machine learning models. The distribution of categories in the gold-standard dataset can be found in the Table 2.

For the purposes of our experiments, the dataset is split into a 70/15/15 division for training, testing, and validation. We use splits with fixed random seeds for reproducibility: 24 for the first split and 42 for the second split. We also ensure that every MD category is present in the validation and test set by re-splitting with a different fixed seed until this is the case.

3.1 Models

As a baseline model, we use the SpanCategorizer (spaCy, 2024) pipeline in the Python library spaCy. This consists of a Tok2Vec base layer to vectorize tokens followed by the SpanCategorizer, which suggests possible spans and then assigns probability values to all possible class labels for those spans. Spans with class label probabilities above a given threshold are identified as spans and categorized. The evaluation of this baseline model can be done within Prodigy.

Label	Frequency	Percentage
RPR	22	0.05%
RFL	39	0.08%
CMT	194	0.42%
CLF	306	0.66%
EXP	524	1.14%
MNT	680	1.47%
ORS	21706	47.07%
PED	458	0.99%
ENM	2135	4.63%
EDP	58	0.13%
PRV	144	0.31%
RVW	206	0.45%
EPA	5979	12.97%
HDG	6060	13.14%
BST	5753	12.48%
SAL	260	0.56%
MNC	368	0.80%
MCD	749	1.62%
AAR	101	0.22%
MNM	59	0.13%
IMS	53	0.11%
MIS	57	0.12%
ERR	196	0.43%
UCT	8	0.02%

Table 2: Frequency and percentage of each label

Our other models are based on the available pretrained SpanBERT models, base-cased and large-cased. We used cased models to facilitate the identification of sentence boundaries. We then tune these models on the task of MD classification using the training portion of our corpus.

We use SpanBERT to jointly perform the identification and classification tasks. Predicting a span where no span was annotated is considered an incorrect prediction. However, this means we need to provide negative examples in training, i.e. examples of things that are not a metadiscourse span. We evaluate three strategies for this:

Token With token negative examples, every token that is not covered by an annotated MD span in the training data, is considered an unlabeled (category '-') span in the training data. This is a

simple strategy but the fact that every negatively labeled span is 1 token in length may cause issues. It also clashes with the pre-training objective of SpanBERT, where training examples are longer spans. Another potential issue is that the overwhelming majority of training examples will be negatively labeled spans with this approach. Note that we are referring here to tokens before subword tokenization, so all spans are whole words.

Spans With spans as negative examples, we group every token that is not covered by an annotated MD span into negatively labeled spans. We approximate the same distribution of span lengths that SpanBERT was pre-trained on: for each span, the length is drawn from a geometric distribution $l\ Geo(0.2)$, which is skewed towards shorter spans, with l_max of 10 (Joshi et al., 2019). This approach has more realistic span length, but it still has the issue that a large majority of training examples will be negatively labeled spans.

Balanced spans With balanced spans as negative samples, the number of spans is balanced for frequency. In this approach, not every token is part of a span, instead, for every metadiscourse span in a text, we also take a negatively labeled span. This span is chosen from a random position and with a length following the SpanBERT pre-training distribution.

For the SpanBERT models, the hyperparameters per_device_train_batch_size and per_device_eval_batch_size are set to 4 and 16 respectively. We run the models for 4 epochs. We performed hyperparameter tuning for the learning rate, weight decay and number of warmup steps. We found that relatively high learning rates are beneficial here so we chose 4e-5. Relatively high weight decays were also beneficial, and we set this to 0.02. High weight decays reduce overfitting through regularization. Warmup steps had no effect on performance and we set it to 500.

4 Results

Starting with the spaCy baseline, we observe the effects of the class imbalance inherent in our task. As shown in table 3, only the five most frequent labels show non-zero performance metrics, while it does not learn to predict the other labels. The baseline's overall accuracy based solely on the non-zero values achieved 80.75, but drops to 16.82 when all categories are considered. This indicates that the baseline model fails to generalize across all MD categories. In addition, the recall scores are lower than the precision scores, particularly for the ENM, HDG, and BST categories, suggesting that many MD markers are incorrectly omitted by the baseline model. The ERR, MIS and UCT categories are not included in our evaluations because they are labels for improper language uses instead of MD and they did not occur as often as we expected.

Label	Precision	Recall	F1 Score
RPR	0.00	0.00	0.00
RFL	0.00	0.00	0.00
CMT	0.00	0.00	0.00
CLF	0.00	0.00	0.00
EXP	0.00	0.00	0.00
MNT	0.00	0.00	0.00
ORS	88.51	88.02	88.26
PED	0.00	0.00	0.00
ENM	95.50	44.16	60.39
EDP	0.00	0.00	0.00
PRV	0.00	0.00	0.00
RVW	0.00	0.00	0.00
EPA	91.20	89.52	90.35
HDG	86.99	72.91	79.33
BST	88.40	69.70	77.94
SAL	0.00	0.00	0.00
MNC	0.00	0.00	0.00
MCD	0.00	0.00	0.00
AAR	0.00	0.00	0.00
MNM	0.00	0.00	0.00
IMS	0.00	0.00	0.00
MIS	0.00	0.00	0.00
ERR	0.00	0.00	0.00
UCT	0.00	0.00	0.00

Table 3: Classification accuracy of the baseline

4.1 SpanBERT

For the SpanBERT models, the performance under different conditions is summarized in Table 4. We can see that surprisingly, the negative sampling strategy of adding every token as a one-word negatively labeled span performed best. Tuning the large pretrained model on MD classification slightly outperformed using the base model.

Next, we examine the per-class performance in Table 5. The SpanBERT-based model is able to classify more of the classes than the spaCy baseline. Nevertheless, there are still some classes that are not identified and classified correctly, and this

Model	Acc.	WeightF1	MacroF1
base	0.83	0.86	0.49
balancedspans	0.75	0.83	0.57
spans	0.12	0.17	0.04
large	**0.86**	**0.90**	**0.67**

Table 4: Evaluation results for different data divisions

appears to be due to class imbalance – there is a strong correlation with frequency. Figure 4 plots the relationship between the amount of instances in a category (frequency) versus the F1 score to visualize this pattern. We can observe a clear correlation between the two variables. For the categories that occur 0-100 times in the test set, two are not classified at all and three are classified with reasonable performance, similar to some of the middle-frequency categories. For the categories that occur 100-1000 times, we can observe a range of F-scores. The categories that occur more than 1000 times are all classified very well.

This raises the question as to what causes these differences in performance in the middle frequency bracket. One potential explanation is lexical variability – categories that can be expressed by a larger range of words should be more difficult to classify.

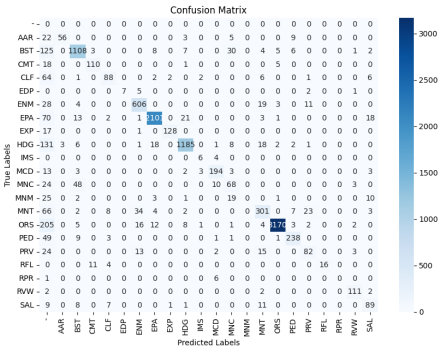

Figure 3: Confusion matrix after tuning spanbert-large for 4 epochs

4.2 Unique spans per category

An interesting property of metadiscourse categories is that some categories have far less lexical variability than others. Textual MD markers are often fixed in form, while interpersonal MD can be expressed in many ways, as discussed in the introduction. Therefore, we also examine the effect of lexical variability on performance per category. Figure 5 plots each category in terms of their ratio of unique spans (lexical variability), controlled for frequency,

Category	P	R	F1	N
RPR	0.00	0.00	0.00	7
RFL	1.00	0.52	0.68	31
CMT	0.89	0.82	0.85	134
CLF	0.79	0.51	0.62	172
EXP	0.98	0.88	0.92	146
MNT	0.79	0.67	0.72	450
ORS	0.99	0.92	0.96	3429
PED	0.90	0.79	0.84	302
ENM	0.90	0.90	0.90	671
EDP	1.00	0.47	0.64	15
PRV	0.67	0.59	0.63	139
RVW	0.92	0.95	0.93	117
EPA	0.98	0.94	0.96	2230
HDG	0.96	0.86	0.91	1376
BST	0.92	0.85	0.88	1299
SAL	0.67	0.71	0.69	126
MNC	0.50	0.44	0.47	153
MCD	0.89	0.88	0.88	221
AAR	0.95	0.59	0.73	95
MNM	0.00	0.00	0.00	61
IMS	0.50	0.60	0.55	10
Acc.			0.86	11184
Macro	0.74	0.63	0.67	11184
Weighted	0.93	0.86	0.90	11184

Table 5: Classification report large cased model

against the F1 score. Frequency is controlled by dividing the number of unique spans by the total amount of spans of that category, similar to how type/token ratio is computed. So, for example, the *Anticipation* (ARR) category has a ratio of 1, meaning that every instance of it uses a unique sequence of tokens.

In the figure, we can see that lexical variability in a MD category does not have to lead to poor performance, although there is a trend that the most variable categories have somewhat lower F-scores. The *Reviewing* (RVW) category shows that even a highly lexically variable category can still get a high F-score.

It is expected that the more lexically diverse categories are more difficult to classify, but the EPA category shows that having almost double the amount of unique spans does not lead to a lower performance than the other big categories. Some of the categories with a ratio between 0.9 and 1 perform similar to categories with a uniqueness ratio of 0.6-0.7. This indicates that the model has good generalization capabilities and needs to do more

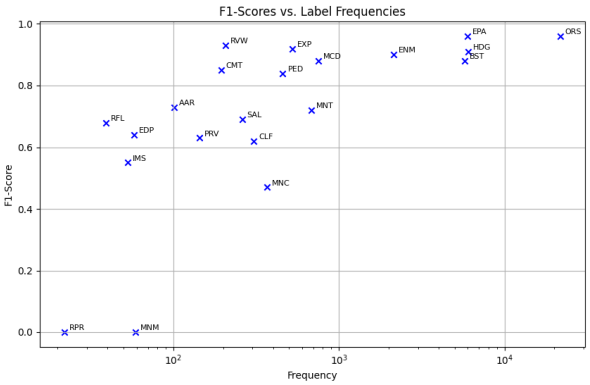

Figure 4: Relationship between frequency and F1 score

than remembering some common words for each category. We also note a pattern of more frequent categories having a lower lexical variability ratio – this is tied to the frequency factor, as among more spans there are more likely to be repeated ones.

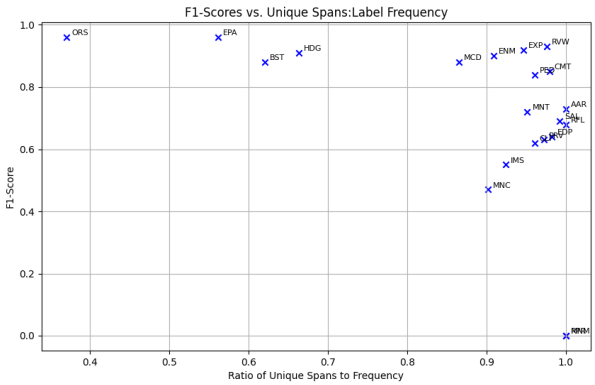

Figure 5: Relationship between the number of unique spans per category divided by frequency and F1 score

4.3 Class ambiguity

Table 6 contains an overview of the most common misclassifications, highlighting elements from Figure 3. Such classification mixups indicate ambiguity between the categories, possibly because of similar MD spans. This overview excludes misidentifications, where an annotated span is falsely identified as not being a metadiscourse span.

The first six categories are the relatively most commonly misclassified categories, and the correlation with category frequency is clearly visible. For the least frequent class, *repairing* (RPR), we see that 6 of 7 instances are misclassified as *managing audience discipline* (MCD, the remaining one is misidentified). The bottom three categories are the three biggest categories. We can see that the most misclassified categories are not misclassified as highly frequent categories, but as other uncommon categories, except *managing comprehension* (MNC) which gets confused for *boosters* (BST, 1299 instances). So, these confusions are more than a frequency effect. The three biggest categories are also not mainly confused with other big categories. This ambiguity instead appears to be caused by frequent overlapping of the two labels. They may co-occur in the same span a lot, as in our Example 2, where a *booster* (BST) occurs inside a *epistemic attitude* (EPA) indication. Another potential cause of confusion is semantic ambiguity and similarity. As an example, for the the ORS-ENM confusion (*Organizing Statements-Enumerating*), we observe that ORS spans that get misclassified as ENM often contain the word 'following', which is also used to enumerate.

Category	Most misclassified	Frequency
RPR	MCD	6/7
IMS	MCD	4/10
RFL	CMT	11/31
EDP	ENM	5/15
MNC	BST	48/153
MNM	MNC	19/61
HDG	MNT	18/1376
EPA	HDG	21/2230
ORS	ENM	16/3429

Table 6: Misclassification analysis of categories

5 Discussion

Overall, we observed that SpanBERT-based approaches are able to perform metadiscourse identification and classification well, clearly outperforming the spaCy baseline (F1-score of 0.90 vs 0.77) and learning to classify more MD categories than spaCy is able to. The best performance was reached with a hyperparameter-tuned version of SpanBERT-large. We tested different strategies for providing negative examples of non-metadiscourse span, but the simplest solution of providing every non-annotated token as a one-word span performed best. Hyperparameter tuning only had a small impact on performance.

The method has promising applications. First, we can use this method to expand our current corpus, which is still under development, to overcome the research limitations imposed by the small amount of available data. Moreover, automatic MD classification is beneficial to language acquisition research, particularly for second language acquisi-

tion. It can help with tasks related to (automated) language assessment, language teaching. It also has potential to be used in text analysis tools to provide language learners with concrete feedback on language coherence and interactionality. MD classification also provides an indication of the viability of the annotation of similar pragmatic and discursive properties of texts, in other words, linguistic items that are also highly context-dependent and potentially challenging even to state-of-the-art NLP methods. The integration of explicit MD representation has the potential to improve chatbots' performance by making their language sound more human-like and making them more effectively understand the users' requests.

The practical applicability of our results is limited by the poor performance on low-frequency classes. In future work, targeted supplemental annotation to make sure all categories have a significant number of instances is the most promising next step. Another future research direction is the exploration of more LLM options. Other types of bidirectional LLMs, such as RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020), can also be fine-tuned on a span identification and classification objective. It is less obvious how to apply autoregressive LLMs to the task of metadiscourse detection, but given their recent success, this would also be a worthwhile direction to explore.

Lastly, it has not become entirely clear how the negative examples in tuning affect model performance. Contrary to our expectations, using more span-like negative examples did not improve model performance. It would be interesting to apply explainable AI techniques to learn more about how spans are represented in SpanBERT and how they are affected by tuning. For example, the model might have a bias towards short spans, which would partially account for our results. Another option is to try to make the negative example spans more natural, e.g. by selecting linguistic constituents as spans rather than random chunks of tokens, to potentially improve performance.

6 Conclusions

The model discussed in this paper achieved an accuracy and F1 score of 0.86 and 0.90 respectively on the task of metadiscourse identification and annotation, representing an improvement of the baseline of 0.77 accuracy and 0.77 F1 score. Obtaining this performance with a fairly small annotated dataset shows that LLMs have potential to help speed up the annotation process even for fairly uncommon NLP tasks such as ours. The main open issue with this task is class imbalance. Two of the smallest categories completely fail to classify. Annotating more conversations where these categories are present is the most promising approach to improve this. Other techniques such as data augmentation, upsampling, and synthetic data generation might also be used. The six biggest categories perform very well, and account for 88.48% of the data.

7 Limitations

The main limitation of this project is the insufficient number of instances for the majority of categories. Annotating more conversations to make sure all categories have a significant number of instances is the most promising next step, as mentioned earlier.

Another limitation is that we were not able to find out why models that are trained on many non-MD spans perform better. Usually we would expect that this causes a larger class imbalance and deteriorated performance. This suggests that there might be better strategies of providing the models with negative training examples.

The applicability of the method is limited by the fact that it requires tuning on metadiscourse annotation, a type of annotation that is not commonly available. This limits the extent to which our method can be applied in diverse linguistic contexts and domain contexts. With our findings, semi-automatic annotation of more data can be explored, but only for languages where usable MD annotation already exists.

We investigated the effect of lexical variability, but our approach has some limitations. For example, if two instances of spans use the same words but in different order, or the same sequence of words but with one word omitted, they would count as different unique spans. Some sort of overlap or semantic similarity-based metric might give a better view of uniqueness of spans.

Due to resource constraints, models were trained for a limited number of epochs. Better model performance might be achieved with more extensive training.

References

Annelie Ädel. 2010. Just to give you kind of a map of where we are going: A taxonomy of metadiscourse in

spoken and written academic english. *Nordic Journal of English Studies*, 9(2):69–97.

Ghada Alharbi. 2016. Metadiscourse tagging in academic lectures.

Avon Crismore, Raija Markkanen, and Margrat S. Steffensen. 1993. Metadiscourse in persuasive writing: A study of texts written by american and finnish university students. *Written Communication*, 10(1):39–71.

Rui Pedro dos Santos Correia. 2013. Automatic classification of metadiscourse for presentation skills instruction. Ph.D. Thesis Proposal.

Rui Pedro dos Santos Correia. 2018. *Automatic Classification of Metadiscourse*. Ph.D. thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

M.A.K. Halliday. 1994. *An Introduction to Functional Grammar*. Hodder Arnold.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Matthew Honnibal, Ines Montani, et al. 2024. Prodigy: An annotation tool for AI, Machine Learning & NLP. https://prodi.gy.

Ken Hyland. 2005. Metadiscourse: Exploring interaction in writing. *Continuum*.

Shin'ichiro Ishikawa. 2023. *The ICNALE Guide: An Introduction to a Learner Corpus Study on Asian Learners' L2 English*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans.

William Vande Kopple. 1985. Some exploratory discourse on metadiscourse. *College Composition & Communication*, 36(1):82–93.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Anna Mauranen. 1993. Cultural differences in academic discourse–problems of a linguistic and cultural minority. *AFinLAn vuosikirja*, pages 157–174.

ED Milne. 2003. Metadiscourse revisited: a contrastive study of persuasive writing in professional discourse. regreso al metadiscurso: estudio contrastivo de la persuasión en el discurso profesional. *Estudios ingleses de la Universidad Complutense*, 11:29–52.

spaCy. 2024. Span categorizer: Pipeline component. https://spacy.io/api/spancategorizer. Accessed: 2024-10-14.

Chengwei Wei, Yun-Cheng Wang, Bin Wang, and C.-C. Jay Kuo. 2023. An overview of language models: Recent developments and outlook. Chapter 5.

Yonghui Wu. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Annelie Ädel. 2006. *Metadiscourse in L1 and L2 English*.

A Software specifications

Python: 3.11.6
json: 2.0.9
numpy: 1.26.1
torch: 2.1.1
scikit-learn: 1.3.2
matplotlib: 3.8.2
seaborn: 0.13.0
transformers: 4.35.0

This project primarily employs the 'SpanBERT/spanbert-base-cased' model, but also the 'SpanBERT/spanbert-large-cased' model, as available on Hugging Face. The specific versions corresponds to the final commits made on May 19, 2021.

The code is attached as supplementary materials and will be on GitHub. The dataset is available upon request, as it was collected for another study that has not been published yet. Interested parties may contact the author directly to obtain access.

B Hardware specifications

Chip: Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz
Total Number of Cores: 48
Memory: 188 GB
Each model training run for our SpanBERT-base models takes about 50 minutes on this hardware, or about 3 hours on a regular desktop PC. 16GB of memory is sufficient.