Classificação de publicações com conteúdo revelatório em uma comunidade do *Reddit*

Artur E. L. e Cupelli[1], Maria Eduarda R. Garcia[1], Mirela Mei[1], Raquel Cristiane da S. Almeida[1]

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

{artur.cupelli,mariaedrgarcia,mirelameic,raquelcsa}@usp.br

Abstract. *The present research addresses the classification of posts with revealing content in a specific community on Reddit, i.e. the detection of spoilers in the subreddit dedicated to the TV show RuPaul's Drag Race. The goal is to understand whether the creation of a classifier for a more specific dataset can achieve better metrics. Utilizing machine learning techniques, especially text classification, and NLP applied to the dataset consisting of posts labeled as spoilers and non-spoilers, the results obtained through the application of different approaches already tested by authors in the reviewed article were analyzed.*
Keywords: *Spoiler classification; Machine Learning; Natural Language Processing; Reddit.*

Resumo. *A presente pesquisa aborda a classificação de publicações com conteúdo revelatório em uma comunidade específica do Reddit, isto é, a detecção de spoilers no subreddit dedicado ao programa de TV RuPaul's Drag Race, visando entender se a criação de um classificador para um conjunto de dados mais específico pode alcançar métricas melhores. Utilizando técnicas de aprendizado de máquina, especialmente classificação de texto, e processamento de linguagem natural, aplicadas sobre o conjunto de dados composto por postagens rotuladas como spoilers e não-spoilers, foram analizados os resultados obtidos através da aplicação de diferentes abordagens já testadas por autores dos artigos revisados.*
Palavras-chave: *Classificação de spoilers; Aprendizado de Máquina; Processamento de Linguagem Natural; Reddit.*

1. Introdução

A presença de informações revelatórias (*spoilers*) em postagens de redes sociais pode causar enorme frustração aos usuários desavisados, diminuindo a excitação da experiência inédita. Isso ocorre visto que a curiosidade e a incerteza não são mais exploradas durante a experimentação de um filme ou livro, por exemplo [Boyd-Graber et al. 2013].

Portanto, um dos principais desafios desses usuários é encontrar maneiras de evitar contato com tais informações indesejadas. Porém, este desafio não é tão trivial. O imediatismo com que publicações são escritas logo após, ou durante, o lançamento de novos conteúdos dificulta a filtragem de mensagens que possam conter *spoilers*.

A complexidade da questão ainda pode ser aumentada quando se leva em consideração redes sociais nichadas em tópicos mais específicos, como fãs-clubes. Isto,

já que é mais alta a probabilidade de serem abordados temas indesejados dentro deste assunto de interesse. O *Reddit*[1] é um exemplo de rede social baseada em comunidades, sendo comum a participação dos usuários em grupos sobre programas, séries, filmes ou livros de seu interesse.

Apesar de diversas comunidades do *Reddit* apontarem a necessidade de avisos de *spoiler* em suas publicações, é comum usuários esquecerem desta regra e acabarem por submeter textos inadequados. Quando isto acontece é necessários que moderadores da comunidade, acionados por denúncias, excluam a publicação. Mas, até que o movimento de punição ocorra muitos usuários podem já ter sido chateados pelas informações revelatórias.

O objetivo da pesquisa é criar um mecanismo de detecção automática e instantânea de *spoilers* em publicações de uma comunidade do *Reddit*, para facilitar a revisão dos moderadores e evitar que muitos usuários entrem em contato com submissões revelatórias. Deste modo, justifica-se a importância da pesquisa em auxiliar participantes da comunidade a conseguir navegar e interagir mais tranquilamente durante lançamentos de novos conteúdos.

Também, visa-se entender se a tarefa de identificação de *spoilers* alcança melhor desempenho em contextos mais fechados, visto que uma comunidade terá um escopo fechado que guia as publicações. Outra hipótese que deseja ser validada é se o uso de informações sobre o nível de interação do autor com a comunidade ajuda ou prejudica o desempenho do modelo de detecção.

Neste contexto, considera-se "detecção" como sinônimo de "classificação", um dos grupos de problemas clássicos da mineração de textos. Também serão aplicadas técnicas de processamento de linguagem natural para tratamentos dos textos das publicações, além de mapeamento de outros metadados sobre as submissões.

Ao decorrer do artigo, serão revisados alguns dos trabalhos relevantes no assunto de detecção de *spoilers*, levantando técnicas já validadas além das métricas obtidas para posterior comparação com os resultados obtidos.

2. Conceitos Básicos

2.1. *Spoilers*

Um *spoiler* pode ser definido como uma informação sobre um programa de TV que afete ou estrague (como na tradução literal do termo) a experiência do telespectador ao revelar-lhe um ou mais fatos importantes acerca do programa que ainda não eram de seu conhecimento [Jeon et al. 2016].

Para além da televisão, *spoilers* podem estar relacionados a diversos outros contextos, e podem ser obtidos através de diversos canais, sobretudo através das redes sociais. Quando se trata de programas de TV, caso fosse possível que todos os telespectadores assistissem ao programa concomitantemente, *spoilers* não seriam um problema, mas inúmeros motivos tornam tal feito impossível. Assim, ainda de acordo com Jeon et al. [2016], não é incomum que usuários isolem-se do convívio nas redes para evitar *spoilers* sobre conteúdos que ainda não consumiram, o que é algo que torna-se cada vez mais

[1] www.reddit.com

difícil tendo em vista a constante presença e necessidade das redes sociais no cotidiano de muitas pessoas.

2.2. *Reddit*

De acordo com a *Wikipedia*[2], o Reddit é um ”agregador social de notícias”. No entanto, aqui opta-se por defini-lo como uma rede social baseada em comunidades. Nele, os usuários podem participar de diversas comunidades – geralmente nichadas por conteúdo – conhecidas como *subreddits*, nas quais ocorre o compartilhamento de informações, notícias, discussões etc. curadas pelos próprios usuários. Cada *subreddit* é moderado por usuários voluntários, que são responsáveis pela manutenção das regras e diretrizes pertencentes a cada comunidade.

Atualmente, o Reddit conta com mais de 50 milhões de usuários ativos diariamente, e é conhecido pela diversidade de conteúdo e perspectivas presentes em suas comunidades.

2.3. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (ou NLP – *Natural Language Processing*) é a área da Inteligência Artificial que lida com a representação e manipulação de linguagem humana, que nesta pesquisa está presente em publicações feitas por usuários do *Reddit*. Entender, classificar, processar e gerar novos textos a partir de dados pré-existentes são alguns dos principais objetivos desse campo, que para atingir tais objetivos utiliza uma variedade de técnicas e tarefas relacionadas ao processamento da linguagem [Lindo 2020].

Técnicas como *stemming* e *lemmatization* [Lindo 2020], que são técnicas de pré-processamento de texto usadas para reduzir as palavras à sua forma básica ou raiz a fim de facilitar a análise, também são empregadas no processamento de textos, bem como as formas de representações *Bag of Words* (Saco de Palavras) e TF-IDF (*Term Frequency-Inverse Document Frequency*) para a transformação de documentos em formatos compreensiveis para os modelos de aprendizado de máquina [Lindo 2020].

2.4. Classificação de Textos

A tarefa de classificação de textos é uma das aplicações mais comuns do NLP, e consiste no aprendizado de rótulos ou categorias a um conjunto de dados tendo seu conteúdo textual como base. A entrada do algoritmo será um vetor de algum tipo de representação do texto, como os citados anteriormente.

Alguns algoritmos clássicos para a tarefa de classificação são: *Naive Bayes*, SVM (*Support Vector Machines*), KNN (*K-Nearest Neighbors*) e redes neurais [Lindo 2020].

3. Trabalhos Correlatos

Os artigos foram levantados para revisão considerando os dez primeiros resultados, por ordem de relevância, da pesquisa do termo “spoiler detection algorithm” na plataforma *Google Schoolar*[3].

[2]https://pt.wikipedia.org/wiki/Reddit
[3]https://scholar.google.com.br/

Diversas técnicas de classificação já foram testadas e documentadas para o problema de detecção de *spoilers*. Contudo, poucos autores utilizaram dados especificamente de publicações de redes sociais, aproveitando-se de outros tipos de texto mais estruturados e bem escritos. Dentre as principais fontes de dados utilizadas pode-se citar o site *TV Tropes*[4], aplicados por Boyd-Graber et al. [2013] e Wr oblewska et al. [2021]; e o site IMDb (*Internet Movie Database*)[5] citados por Marukatat [2022] e Lindo [2020]. A vantagem de utilizar dados escritos com maior cautela é a escassa presença de jargões e abreviações típicas de redes sociais, como "você" que é comumente abreviado para "vc". Outro exemplo é a repetição de alguns carácteres indicando reação de raiva, animação ou risada (como "aaaaaaa" ou "hahahaha"). Tal característica pode ser uma das explicações pela qual o autor que utilizou dados do *Twitter*[6] obteve menor desempenho (Tabela 1).

Dentre os métodos utilizados para classificação, a maioria das publicações comparou algumas abordagens supervisionadas e indicaram a que obteve melhor desempenho. Boyd-Graber et al. [2013], Marukatat [2022], Jeon et al. [2016] e Hijikata et al. [2016] elegeram o algoritmo *SVM* como melhor abordagem de extração [Boyd-Graber et al. 2013], enquanto Ueno et al. [2019] e Lindo [2020] a rede neural *LSTM*. Na Tabela 1 estão detalhadas as abordagens utilizadas por cada autor, além das métricas obtidas.

Comparando com outras tarefas mais clássicas da classificação de texto, como a categorização de notícias, percebem-se que as métricas divulgadas nas publicações não são exatamente as mais atraentes. Isso corrobora com a complexidade da tarefa de detecção de *spoilers*, visto a subjetividade e temporalidade envolvida na definição do que deve ser considerado revelatório.

[4] *TV Tropes* é um portal que documenta e exemplifica "tropos narrativos" – padrões narrativos repetidos em diversas histórias. Disponível em: www.https://tvtropes.org/

[5] IMDb é uma plataforma que armazena informações sobre filmes, séries e programas televisivos. Disponível em: www.imdb.com

[6] Antes conhecida como *Twitter*, hoje nomeada *X*, é uma rede social para compartilhamento de mensagens curtas e notícias. Disponível em: www.x.com

Nome da publicação	Autores	País dos Autores	Algoritmo de classificação	Métrica	Dados utilizados
Spoiler Alert Machine Learning Approaches To Detect Social Media Posts With Revelatory Information	[Boyd-Graber et al. 2013]	EUA	SVM	0.67 (Acurácia)	TV Tropes
Using Cohesion-Based and Sentiment-Based Attributes to Classify Spoilers in Movie Reviews	[Marukatat 2022]	Tailândia	SVM	0.78 (Acurácia)	IMDb
Spoiler Detection and Early Spoiler Avoidance	[Umaji et al.]	Índia	*	*	*
A Basic Study On Spoiler Detection From Review Comments Using Story Documents	[Maeda et al. 2016]	Japão	Keyword Matching	Não cita	”Story Documents”Japoneses
Spoiler Detection In Tv Program Tweets	[Jeon et al. 2016]	Coreia do Sul	SVM	0.79 (F-Score)	Twitter
Spoiler In A Textstack How Much Can Transformers Help	[Wróblewska et al. 2021]	Polônia	Transformer	0.88 (ROC AUC)	TV Tropes
A Spoiler Detection Method For Japanese-written Reviews Of Stories	[Ueno et al. 2019]	Japão	LSTM	0.55 (F-Score)	Rakuten Shopping Mall
Context-based Plot Detection From Online Review Comments For Preventing Spoilers	[Hijikata et al. 2016]	Japão	SVM	0.77 (F-Score)	Amazon
Sentence-Based Plot Classification for Online Review Comments	[Iwai et al. 2014]	Japão	Naive Bayes	0.77 (F-Score)	Amazon
Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings	[Lindo 2020]	Irlanda	LSTM	0.86 (F-Score)	IMDb

Tabela 1. Mapeamento das abordagens, dados e métricas obtidas pelas publicações revisadas

Boyd-Graber et al. [2013] levantam um experimento indicando que o uso de metadados atuando como *features* adicionais em um modelo de classificação ajudam a aumentar sua performance. O autor testa diferentes combinações de dados adicionais e comprova que eles podem adicionar informações revelantes ao algoritmo que acaba obtendo métricas melhores.

Pode-se ressaltar que nenhum dos artigos revisados utilizou dados da rede social *Reddit*, o que será uma nova perspectiva desta pesquisa. Também, foi notado que as publicações focam no problema de identificação de *spoilers* em um contexto geral, ou seja, não se preocupam em saber se um dado texto contém revelações sobre a franquia “Senhor dos Anéis”, por exemplo, mas sim de qualquer conteúdo.

4. Experimento

Para implementar o modelo de aprendizado supervisionado, foram coletados dados das 7903 publicações mais recentes da comunidade de *Rupaul's Drag Race* do *Reddit*. A comunidade foi escolhida pela quantidade de membros e publicações, além da obrigatoriedade do alerta de *spoilers* nas submissões. Ou seja, um dos metadados da publicação é um campo binário “*is_spoiler*”, sendo justamente o rótulo que será aprendido pelo modelo.

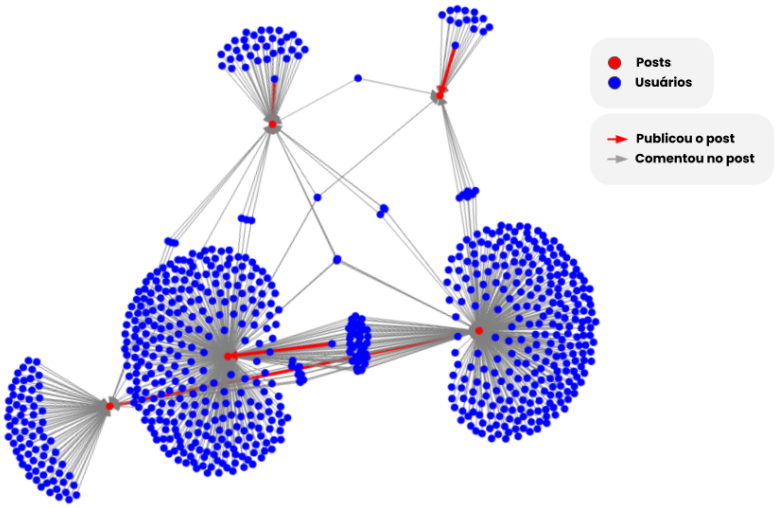

Figura 1. Exemplo do grafo G_5 construído com as 5 primeiras publicações da base

4.1. Coleta dos dados

A coleta foi realizada a partir de requisições *HTTP* enviadas à *API* do *Reddit*, com uma chave de autenticação gerada pela plataforma de desenvolvedores da rede[7].

Para cada publicação, foram coletados os seguintes atributos: título; texto; identificador; autor; se é uma publicação com *spoiler*; data da submissão; e uma lista dos usuários que comentaram. A última informação foi coletada para construção de um grafo de relações da comunidade.

4.2. Construção do grafo de relações da comunidade

Uma dos objetivos da pesquisa é entender se o nível de engajamento do autor pode indicar maior ou menor probabilidade de que sua nova publicação contém um *spoiler*. Esta medida será calculada a partir de uma rede de relações.

Dadas as 7903 submissões, foi construído um grafo em que cada nó pode ser uma publicação (vermelho) ou um usuário (azul). Cada aresta direcionada conecta um usuário a uma ou mais publicações caso ele tenha comentado (peso 1) ou seja o autor dela (peso 2). O grafo resultante G possui 72731 nós e 439414 arestas. Visto a impossibilidade da visualização de uma estrutura tão complexa, foram construídos grafos menores G_5 e G_{20} somente para exemplificação visual do resultado (Figuras 1 e 2).

A partir do grafo global construído, para cada autor foi calculado o número de arestas de saída multiplicadas pelo seu peso correspondente. Posteriormente, esta informação foi unida a base principal das publicações, fazendo a união pelo nome do autor.

[7]Para mais informações sobre o processo de cadastro e autorização para utilização dos dados do *Reddit* confira https://www.reddit.com/wiki/api/

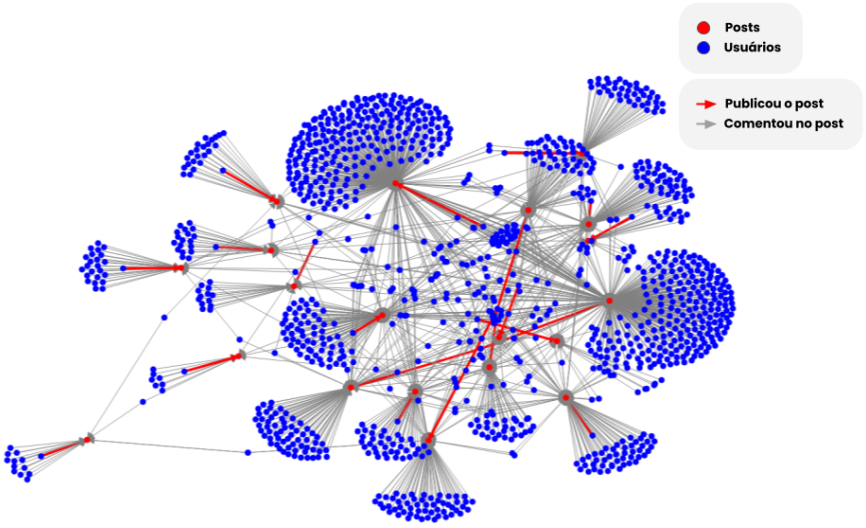

Figura 2. Exemplo do grafo G_{20} construído com as 20 primeiras publicações da base

4.3. Pré-processamento

Para a nova *feature* construída, foi necessária a normalização, visto que pode assumir valores no intervalo de 0 a $número\ de\ publicações\ totais$.

Devido os textos de redes sociais serem extremamente ruidosos, foram aplicadas algumas expressões regulares para remoção de sujeiras conhecidas como hiperlinks, valores numéricos, *emojis* e caracteres especiais. Também foi feita a padronização os textos com letras minúsculas.

O metadado "data da criação" da publicação foi manipulado por *enconding* criando a binarização por dia da semana. Esse processamento adicionou sete novas *features*, que podem ajudar a indicar maior probabilidade de uma publicação conter *spoilers* caso feita em algum dia específico da semana.

Classe	**Conjunto de Treino**		**Conjunto de Teste**
	Antes do balanceamento	**Após balanceamento**	
"Spoiler"	909	4623	425
"Não Spoiler"	4623	4623	1946
Total	5532 (70%)	9246	2371 (30%)

Tabela 2. Quantidade de dados por classe e conjunto

Além disto, a base de dados foi separada em conjuntos de treino (70%) e teste (30%) para aprendizado dos modelos por seleção aleatória. Para o primeiro conjunto,

as classes ("*spoiler*" e "não *spoiler*") foram balanceadas com a estratégia da repetição aleatória da classe minoritária.

4.4. Aplicação do processamento de linguagem natural

Para os atributos "título" e "texto" da publicação, é preciso que seus valores sejam gerenciados por técnicas da NLP. Neste contexto ambos foram considerados como um só texto, portanto o primeiro passo foi concatenar ambos atributos, criando um terceiro resultante.

O vocabulário do modelo foi criado a partir dos unigramas e bigramas do conjunto de treino, com remoção das *stopwords* e limitação dos 150.000 *tokens* mais relevantes. Para cada publicação foi criado um vetor com a representação *TF-IDF*, que apresentou melhor desempenho do que o experimento feito com a *BoG*.

4.5. Construção dos classificadores

Assim como os autores revisados, foram experimentados nesta pesquisa diferentes algoritmos de classifica a fim de comparação. Selecionaram-se três: *SVM*, *KNN* e Regressão Logística.

Para cada um dos classificadores foram realizados três experimentos que variavam a quantidade de *features*:

1. E_1 : Somente vetor de representação do texto
2. E_2 : Vetor de representação do texto + Número de arestas de saída do autor
3. E_3 : Vetor de representação do texto + Número de arestas de saída do autor + Dia da semana da publicação

5. Resultados

A avaliação do desempenho de cada experimento foi extraída pelas métricas: *F-Score*, *Precision*, *Recall* e Acurácia. Porém, para medida final comparativa entre os modelos foi elegida a *F-Score*, devido utilização pelos demais autores revisados. A Tabela 3 apresenta os resultados detalhados.

Algoritmo	Experimento	Acurácia	F-Score	Precision	Recall
SVM	E_1	0.85	**0.39**	0.78	0.26
	E_2	0.82	**0.39**	0.78	0.26
	E_3	0.86	**0.44**	0.77	0.31
KNN	E_1	0.72	**0.44**	0.34	0.61
	E_2	0.78	**0.41**	0.40	0.43
	E_3	0.68	**0.41**	0.30	0.62
Log. Reg.	E_1	0.86	**0.51**	0.71	0.40
	E_2	0.85	**0.53**	0.64	0.45
	E_3	0.86	**0.59**	0.53	0.67

Tabela 3. Métricas obtidas com os experimentos

O algoritmo *SVM*, apesar de muito citado por outros autores como a melhor técnica, apresentou o pior desempenho para este conjunto de dados. Somente a adição

da informação sobre o dia da semana no experimento E_3 indicou aumento significativo no *F-Score* e Acurácia.

Por outro lado o *KNN*, mesmo com a menor quantidade de dados do E_1 já iguala seu desempenho ao E_3 do *SVM*. Porém, a adição de novas *features* nos experimentos seguintes diminui a performance do modelo. É importante ressaltar que este algoritmo é sensível ao hiper parâmetro k, sendo que possíveis refinos na sua escolha com a mudança das *features* poderia ter incentivado melhores métricas.

A regressão logística apresentou o melhor comportamento para o conjunto de dados utilizado. Mesmo nos experimentos E_1 e E_2 o *F-Score* é mais significativo que os demais algoritmos. O último experimento bate 0.59, métrica que até supera um dos autores revisados (0.55 do *LSTM* treinado por [Ueno et al. 2019]). É possível observar maior relevância dentre os experimentos da adição do dado sobre o dia da semana, tanto no *F-Score*, quanto na Acurácia. Por outro lado, o atributo sobre o número de arestas de saída do autor do modelo, não traz conhecimento suficiente ao modelo, visto a baixa alteração na métrica.

6. Conclusão

Os resultados obtidos através da presente pesquisa acerca da classificação de *spoilers* na comunidade de *RuPaul's Drag Race* do *Reddit* indicam que, pelas métricas dos modelos, a influência do número de arestas do autor da publicação – isto é, a quantidade de interações do autor dentro da comunidade – traz certo nível de informação relevante para a classificação, apesar de modesta. Ademais, também é possível observar que a utilização de dados nichados não necessariamente colabora positivamente com a tarefa de classificação de textos com *spoilers*, visto que, de forma geral, o desempenho dos algoritmos neste conjunto de dados foi inferior a maioria dos artigos revisados.

Por fim, é importante ressaltar como uma das contribuições da pesquisa a disponibilização de uma base de dados com diversos metadados que podem ser utilizados por pesquisadores interessados em abordar temas semelhantes ou correlacionados, base essa que pode ser futuramente refinada através da aplicação de outras técnicas de pré-processamento de texto e coleta de mais dados.

Referências

Boyd-Graber, J., Glasgow, K., and Zajac, J. S. (2013). Spoiler alert: Machine learning approaches to detect social media posts with revelatory information. *Proceedings of the American Society for Information Science and Technology*, 50(1):1–9.

Hijikata, Y., Iwai, H., and Nishida, S. (2016). Context-based plot detection from online review comments for preventing spoilers. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 57–65. IEEE.

Iwai, H., Hijikata, Y., Ikeda, K., and Nishida, S. (2014). Sentence-based plot classification for online review comments. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 245–253. IEEE.

Jeon, S., Kim, S., and Yu, H. (2016). Spoiler detection in tv program tweets. *Information Sciences*, 329:220–235.

Lindo, A. (2020). *Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings*. PhD thesis, Dublin, National College of Ireland.

Maeda, K., Hijikata, Y., and Nakamura, S. (2016). A basic study on spoiler detection from review comments using story documents. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 572–577. IEEE.

Marukatat, R. (2022). Using cohesion-based and sentiment-based attributes to classify spoilers in movie reviews. In *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, pages 80–84. IEEE.

Ueno, A., Kamoda, Y., and Takubo, T. (2019). A spoiler detection method for japanese-written reviews of stories. *International Journal of Innovative Computing Information and Control*, 15(1):189–198.

Umaji, K., Mutange, V., Ramana, A., Chayal, S., and Sharma, S. Spoiler detection & early spoiler avoidance.

Wróblewska, A., Rzepiński, P., and Sysko-Romańczuk, S. (2021). Spoiler in a textstack: How much can transformers help? *arXiv preprint arXiv:2112.12913*.