Automatic Segmentation of Narrative Text Into Scenes According to SceneML

Tarfah Alrashid[1], Robert Gaizauskas[2]

[1]University of Jeddah, Jeddah, Saudi Arabia
[2]University of Sheffield, Sheffield, UK

Abstract

Automatically segmenting narrative text into scenes is a complex task that remains relatively underexplored. Scenes form fundamental structural units within narratives, marking shifts in time, location, and character interactions. In this paper, we introduce a supervised learning approach to scene segmentation, using SceneML, an annotation framework for narrative text. We evaluate multiple models, including BERT-based classifiers and Conditional Random Fields (CRF), treating scene segmentation as a sentence classification and sequence labeling task. Our experiments show that the BERT cased model achieves the highest balanced accuracy of 0.58 and an F1 score of 0.24 for the minority class. However, statistical tests revealed no significant differences among BERT-based models but highlighted distinctions between CRF models and BERT models. These results indicate that while supervised learning models can improve scene segmentation, further refinements are needed. We discuss potential enhancements, including sequence-based transformer models, integration of temporal and geographical references, and the investigation of decoder-only models such as GPT-3 and GPT-4. Our findings highlight both the progress and challenges in automating scene segmentation and provide directions for future research.

Keywords

Narrative text, scene segmentation

1. Introduction

Narrative texts, whether in literature, film scripts, or storytelling applications, often follow a structured progression of scenes that convey events, character interactions, and shifts in time and location. Automatically segmenting these texts into scenes can enhance various natural language processing (NLP) tasks such as text summarization, information retrieval, and interactive storytelling. It is also of interest to literary scholars studying variation in narrative structure within and across authors. However, scene segmentation remains a challenging problem due to the complexity of defining and identifying boundaries within a continuous text.

Existing studies on text segmentation primarily focus on topic-based segmentation, lexical cohesion, and discourse structure, but these approaches are insufficient for capturing scene-level transitions in narratives. Previous work specifically on the segmentation of narrative texts has investigated lexical cohesion measures, supervised classification, and event boundary detection, yet none of this work is set within a broad framework for annotation of narrative structure and has achieved limited results.

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds.): Proceedings of the Text2Story'25 Workshop, Lucca (Italy), 10-April-2025

ttal-rashid@uj.edu.sa (T. Alrashid); r.gaizauskas@sheffield.ac.uk (R. Gaizauskas)

This paper introduces an approach to scene segmentation based on SceneML, an annotation framework designed for narrative text [1]. We develop and evaluate supervised learning models that leverage contextual and linguistic features to automatically segment narrative texts into coherent scenes. Our work differs from prior studies by incorporating a more comprehensive scene representation, addressing scene transitions, and using machine learning techniques to enhance segmentation performance.

The remainder of this paper is structured as follows: Section 2 reviews related work in text segmentation and scene detection. Section 3 describes the dataset used for training and evaluation. Section 4 presents the models and experimental setup. Section 5 discusses the results and their implications, followed by the conclusion in Section 6.

2. Related Work

Automatic segmentation of narrative text into scenes remains underexplored. Existing studies address related tasks such as lexical cohesion-based segmentation, feature-based segmentation, and event segmentation but lack comprehensive scene annotation frameworks. Kozima and Furugori [2] use what they call a Lexical Cohesion Profile (LCP) to detect scene boundaries through shifts in lexical cohesion, but the approach's reliance on fixed window sizes limits its applicability. Kauchak and Chen [3] frame segmentation as a classification task, using an SVM classifier with lexical and structural features, yet their approach disregards sequential dependencies, leading to potentially inconsistent segment lengths. Event segmentation has also been studied, focusing on narrative shifts in film based on location, character, and time, though this work does not provide computational models applicable to text [4]. Closest to our work, a more recent study by Zehe et al. [5] developed a scene annotation scheme for German narratives and tested unsupervised and supervised segmentation models. However, their definition of scene differs from ours by requiring not only that a scene be a portion of a narrative where location, characters and time are coherent, i.e. do not change, but which centres on a single central action. We do not require this last condition. Overall, their annotation scheme is quite limited and their segmentation model achieves relatively weak performance (F1 = 0.24). Unlike previous work, our approach builds on a more comprehensive annotation framework, SceneML, that captures a broader range of narrative scene dynamics.

3. Data Set

The dataset used for our study – the ScANT corpus [6] – was constructed for the study of narrative structure and is composed of selected chapters from children's stories and adult novels that are no longer protected by copyright. Children's stories were specifically chosen with the expectation that they would exhibit a relatively simple narrative structure. Conversely, adult novels were included to incorporate more complex narratives, posing a greater challenge for automated analysis of narrative structure. There are three sources for the dataset. The first is 'Bunnies from the Future', a middle-grade children's story authored by Joe Corcoran. The second source is 'The Wonderful Wizard of Oz', originally part of the Brown Corpus. Finally, the third source comprises 'Pride and Prejudice', 'A Tale of Two Cities', 'The Adventures of Sherlock

Holmes' and 'The Great Gatsby' obtained from Project Gutenberg. The dataset is annotated with a subset of the SceneML elements proposed in [1], specifically just SCENE, SCENE DESCRIPTION SEGMENT (SDS) and SCENE TRANSITION SEGMENT (STS), along with the more recently added NON-SCENE element. In brief these may be described as follows:

Scene A *scene* is defined as a unit of narrative in which the time, location and principal characters are constant and in which specific events which constitute the narrative are recounted. Any change in these three elements indicates a change in the scene.

Scene Description Segment (SDS) A scene is realised in written forms of narrative through one or more, potentially non-contiguous, *scene description segments* (SDSs), themselves contiguous sequences of sentences all narrating the same scene. The SDS mechanism allows for the relating of one scene in a narrative to be embedded within another, as for example, in flashback or flashforward.

Scene Transition Segment(STS) Some passages describe not one scene or another but rather the transition between scenes. So, one SDS describing a conversation between two characters A and B in location L could be followed by a single sentence "As soon as B had left, A jumped in a taxi and drove to L′". At L′ a new scene might then unfold. The single sentence joining the two SDSs does not belong to the first scene nor to the second. And it does not constitute a scene in its own right, as no narrative-significant action takes place during the time it describes, save the transition of A to a new location. Its sole narrative function is to indicate a transition from one scene to another. Such elements SceneML refers to as *scene transition segments* (STSs).

Non-scene Elements Aside from STS's, other elements are also present in narrative text. These include general philosophising or opinion segments, background information segments, and narrative summary or narrative catchup (e.g. "It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness ..." from Charles Dickens, *A Tale of Two Cities*). These passages serve a variety of functions but do not relate specific, situated events involving protagonists in the story. All such passages SceneML designates as *non-scene elements*.

The ScANT dataset has 2,796 sentences, 55,635 words and 191 SDSs [1].

4. Models

To build a model that can automatically segment narrative text into scenes (SDSs) using machine learning, first, we need to train the model using training data. To make the problem easier for automatic scene segmentation, we treated the task as a sentence classification problem instead of text segmentation, where each sentence is given a tag (i.e. 1 is designated for sentences on the boundary of an SDS, either at the beginning or the end, and 0 otherwise). Scene Transition Segments are not considered as a separate classification task here as their numbers in the annotated data were very small compared to the number of annotated SDSs.

[1]The corpus is free for research purposes and is available from https://doi.org/10.15131/shef.data.21517908.v1.

The machine-learning models were trained and tested using the ScANT corpus. To ensure robust evaluation, stratified 10-fold cross-validation was implemented using the scikit-learn library. This technique splits the data into 10 equally sized folds while preserving the class distribution, allowing the models to be trained and tested on each fold independently. This approach helps in obtaining reliable performance estimates for the models. Notably, the data were not shuffled to preserve sentence order. The following section presents and explains the machine-learning models used for the task.

Three machine-learning models were trained and tested on the annotated data. Then, we compared the models' performances to determine which model is optimal for our task. The following subsections provide a brief description of each of the models.

4.1. Model 1 - The Conditional Random Field (CRF) Model

In the first model, we treated the problem as a sequence-labelling problem, where the order of sentences is significant and the wider textual context of the sentence being labelled is important. Herein, the sequence refers to the ordered sentences of each chapter and their corresponding tags. A CRF model was trained on the training data. For this endeavour, we first extracted the following features:

- Transitioning phrases: This is a binary feature, where if the sentence contains a transitioning phrase, the feature is given a tag of *1* and *0* otherwise. This feature aims to identify transitions between different segments within the text. Transitioning words/phrases (e.g. *later on*, *after*, etc.) are hypothesised to appear more in sentences on the boundaries of a scene.
- Beginning or end of a paragraph: This is also a binary feature, where if the sentence occurs at the beginning or end of a paragraph, the feature is given a tag of *1* and *0* otherwise. This feature aims to capture paragraph-level patterns that might influence the classification of the current sentence.
- End of a chapter (true/false): This binary feature denotes whether the current sentence occurs at the end of a chapter, as the end of a chapter usually indicates the end of a scene.
- Part-of-speech (POS) tags: Incorporating the part-of-speech tags of each word in the current sentence being classified was carried out using spaCy. In addition, POS tags were extracted for the two preceding sentences and the two following sentences.
- Named entity: Each word in the sentence being classified was given a BIO tag. The Named Entity Recognition (NER) function used was implemented by spaCy, using the NER model `en_core_web_md`. Named entities can include names of people, organisations, locations or other specific entities. In addition, the words of the two preceding sentences and the two following sentences were also given named entity tags.
- Contextual information (2 sentences before and after): This feature considers the two sentences preceding and the two sentences following the current sentence. By incorporating neighbouring sentences, the model can capture contextual dependencies and the influence of surrounding information on the classification of the current sentence. This information presented to the model as a set of features. The same set of features extracted from the test sentence is also extracted from the two preceding sentences and the two following.

- Visually descriptive language (VDL): Visually descriptive information as described in [7] is used here as a feature, on the basis of the hypothesis that a scene change is likely to include a description of a new setting. A classifier was developed to classify sentences as (*0, 1,* or *2*), where:
  - *0* tag: not visually descriptive
  - *1* tag: visually descriptive
  - *2* tag: partially visually descriptive

  To assess the effectiveness of the VDL feature on the performance of the CRF classifier, the model was tested twice—once with the VDL feature added to the list of features and once without.

Table 1 presents the results of the two versions of the CRF model. Section 6 summarises and compares the performance results of all models.

4.2. Model 2 - Bidirectional Encoder Representations from Transformers (BERT)

The second model developed is a deep-learning model that uses a pre-trained language model. The ktrain library [8] was utilised to implement the model, with the use of BERT [9] from Hugging Face transformers. Two experiments were conducted on the model to explore the most effective implementation: one with BERT cased and one with BERT uncased. The model was fine tuned using a learning rate of 1.44E-05, with 3 epochs, and maximum length of 128 for `Bert-Cased`, and 5 epochs, and maximum length of 256 for `Bert-Uncased`.

4.3. Model 3 - Sentence Pair Classification with BERT

In an attempt to capture as much context as possible, the task of scene segmentation was also treated as a sentence pair classification task, where the relationship between a current sentence and its surrounding context is assessed. The input consists of a pair: the first-pair part is the current sentence and the second-pair part is the concatenated form of the sentence itself along with the two preceding and two following sentences. This allows the broader context surrounding the current sentence to be considered during classification (see Figure 1). As with model 2, this model was implemented using the ktrain library. To explore different variations, two experiments were conducted using pre-trained BERT models from the Hugging Face model repository: BERT cased and BERT uncased.

The model was fine tuned using a learning rate of 1.44E-05, with 10 epochs, and maximum length of 512 for `Bert-Cased` and `Bert-Uncased`.

5. Results

Table 1 presents the performance results of the six machine-learning models, namely, CRF, CRF(VDL), which refers to CRF models with the VDL feature added, BERT cased, BERT uncased, Sent-Pair Cased, which is a sentence pair classification with a BERT cased model, and Sent-Pair

$$[s_1, s_2, \ldots, s_N]$$
$$\downarrow$$
$$[(s_1, c_1), (s_2, c_2), \ldots, (s_N, c_N)]$$

$$\text{where } c_i = s_{i-2} + s_{i-1} + s_i + s_{i+1} + s_{i+2}$$

Figure 1: Sentence pair input, where c refers to the concatenated form. Input to the classifier is formed by pairing each sentence in the original text with a context, which is the concatenation of the sentence itself along with the two preceding and two following sentences.

Table 1
SDS boundary classification results, (W) refers to weighted average and (M) refers to macro averaging. P refers to Precision, R refers to Recall and Acc means accuracy. Tenfold-cross-validation was carried out on each of the models. The average of the results of the 10 folds is reported here. MCC refers to the most common class classifier (as a base line).

Model	Acc	Balanced Acc	P(W)	R(W)	F1(W)	P(M)	R(M)	F1(M)	F1 class 0	F1 class 1
Bert Cased	0.92	0.58	0.88	0.89	0.87	0.64	0.58	0.59	0.94	**0.24**
Bert Uncased	0.92	0.56	0.87	0.89	0.88	0.61	0.56	0.57	0.94	0.20
Sent-Pair Cased	0.90	0.51	0.86	0.85	0.84	0.56	0.55	0.54	0.91	0.17
Sent-Pair Uncased	0.90	0.55	0.87	0.85	0.84	0.55	0.53	0.51	0.92	0.18
CRF(VDL)	0.90	0.52	0.85	0.88	0.85	0.59	0.52	0.52	0.93	0.12
CRF	0.87	0.52	0.86	0.89	0.86	0.64	0.52	0.53	0.91	0.12
MCC	0.92	0.50	0.84	0.92	0.88	0.46	0.50	0.48	0.96	0.00

Uncased (with BERT base uncased). The models are evaluated using different metrics, including accuracy, balanced accuracy, precision (with both macro and weighted average), recall (with both macro and weighted average) and F1 (with both macro and weighted average and F1 score for each class 0 and 1 independently). Tenfold-cross-validation was used to test each of the six models.

The findings indicate that accuracy alone showed relatively high values across the models (ranging from 0.87 to 0.92). However, since the dataset is highly imbalanced (there are many more 0 tags than 1 tags), accuracy alone is not sufficient to compare between models. We added other metrics that can give a better insight into the performance of models, such as balanced accuracy and F1 for each individual class.

As can be seen, most of the metrics used in testing the models yielded highly similar results, which made it difficult to determine which model performed best. However, if we focus on the two metrics that can be used to reflect the performance of models on imbalanced datasets, the balanced accuracy metric is often considered when one of the classes is a lot larger than the other. The BERT cased model achieved the highest balanced accuracy of 0.58, indicating its ability to handle imbalanced data more than the other models.

In addition, we obtained the F1 score for each class and focused on the results for a minority

class (class 1) that could help elicit an insight on which of the models would perform better in predicting class 1. Similarly, the BERT cased model scored the highest, with a 0.24 F1 score.

Notably, although the BERT cased model achieved the highest scores among all models in terms of balanced accuracy and F1 (for class 1), it is difficult to derive a conclusion as the results for the majority of the models are highly similar. To see whether the differences in our models' performances are significant, we conducted a statistical test on the results, as reported in the following section.

6. Analysis and Discussion

A statistical analysis was conducted to analyse whether there is a significant difference in the performance of the six models. Our null hypothesis is that there is no significant difference in the performance of the six models that were used for scene boundary detection. Of the 10 metrics presented in Table 1, balanced accuracy and F1 for class 1 are the two metrics chosen to do the statistical analysis on. The metric values for each of the 10 folds were used as the data samples for the significance test. A Mann–Whitney U test [10] was then carried out on these performance metrics of the six models. Mann–Whitney test is non-parametric test that does not require normally distributed data and works well with small data sizes, which is the case in our task (10 BA and 10 F1 scores for each classifier).

6.1. Statistical Analysis

Table 2
Pairwise Mann–Whitney p-value results on the models' performance. For each pair, the p-value was calculated on the balanced accuracy results and on F1 for the minority class. The results were reported as BA | F1 in each case.

Model Comparison	Sent Pair-BERT Uncased	Normal-BERT Cased	Normal-BERT Uncased	CRF-VDL	CRF
Sent Pair-Bert Cased	0.3847 | 0.4048	0.1859 | 0.1402	0.3640 | 0.4043	0.7337 | 0.6488	0.8501 | 0.6770
Sent Pair-Bert Uncased	-	0.2123 | 0.2890	0.7336 | 0.8203	0.1212 | 0.1035	0.1859 | 0.1402
Normal-Bert Cased		-	0.6230 | 0.5449	**0.0211** | **0.0309**	**0.0257** | **0.0341**
Normal-Bert Uncased			-	0.1211 | 0.1397	0.2729 | 0.1617
CRF-VDL				-	1.0 | 0.7896

Table 3
Mann-Whitney Test Results for Balanced Accuracy

Comparison	Sent Pair-Bert Cased	Sent Pair-Bert Uncased	Normal-Bert Cased	Normal-Bert Uncased	CRF-VDL	CRF
MCC	0.1153	0.0014	6.39e-05	0.0014	0.0426	0.1153

In general, as shown in Table 2, the results showed no significant difference (p-value> 0.05) in the performance of the models. In terms of balanced accuracy metric, the p-values ranged from 0.1859 to 0.7336, suggesting no significant difference in the performances of BERT cased, BERT uncased, sentence pair with BERT cased and sentence pair with BERT uncased. This is also the case for the results in terms of F1 for class 1. The p-values for the model comparisons ranged from 0.1402 to 0.8203, suggesting no significant difference in the performances of BERT

cased, BERT uncased, sentence pair with BERT cased and sentence pair with BERT uncased. On the other hand, the p-values for both metrics (balanced accuracy and F1 for class 1) showed a significant difference (p-value < 0.05) in the performance of BERT cased compared with CRF and CRF with the VDL feature.

In addition, another statistical analysis was conducted to analyse whether there is a significant difference in the performance of the six models compared to the MCC baseline. Table 3 shows Mann-Whitney p-value results between the most common class (MCC) classifier and each of the six models. These p-values were obtained for 10-fold BA, as the F1 for the minority class will be all 0s for the MCC. The results show that there is a significant difference in the performance between the MCC and sentence pair with Bert Uncased, Bert Cased, and Bert Uncased. There is marginally significant evidence of a difference with CRF-VDL. And finally, there is no strong evidence of a difference with sentence pair with Bert Cased and CRF.

6.2. Discussion

The stronger performance of BERT could be attributed to the fact that BERT is pre-trained on the BookCorpus collected by [11]. The BookCorpus is made of 11,038 novels from 16 different genres (e.g. romance, science fiction, fantasy, etc.). Therefore, BERT has seen narrative text previously.

Overall, the findings suggest that the choice of model (BERT cased, BERT uncased, sentence pair with BERT cased and sentence pair with BERT uncased) may not significantly impact performance in our tasks. Users can select the model that best aligns with their specific requirements or preferences without compromising performance.

However, there is no significant difference both between the CRF models and the some of BERT models and between the two CRF models. This could suggest: (1) the power of language models pretrained on large amounts of text and then fine-tuned for the task outweighs the use of features engineered for this specific task but then trained on a small amount of labelled data (2) VDL either offers no help for this task or the accuracy level of the VDL classification is too low to be useful here.

Finally, comparing the performance of our models to those of Zehe et al. [5] on the binary scene segmentation task they define, we see that our results are broadly similar (0.24 F1 measure). Given differences in task definition and dataset not too much should be made of this without further investigation. However, both their efforts and ours suggest this is indeed a hard task.

7. Conclusion

Among the models evaluated, the BERT cased model achieved the highest performance for scene segmentation, with a balanced accuracy of 0.58 and an F1 score of 0.24 for the minority class. However, statistical analysis using the Mann-Whitney test revealed no significant differences among the BERT-based models, including their cased and uncased versions. Additionally, while there was a significant difference between the CRF models and the BERT cased model, no significant difference was found between the sentence pair BERT cased model and the MCC baseline. Interestingly, a marginally significant difference was observed between MCC and CRF-VDL, whereas no significant difference was found between MCC and the standard CRF

model. These findings highlight the complexity of the scene segmentation task and suggest that while BERT-based models demonstrate improved performance, the differences among approaches may not be substantial.

8. Future Work

Although we have made progress in investigating supervised models for scene segmentation, there is clearly still room for substantial improvement. As an initial step, designating some our corpus as a development data, as distinct from training and test data, would allow us to conduct some failure analysis to determine which cases in particular various models are finding challenging. Of course acquiring more labelled data should also help – how sensitive task performance is to training set size is not known.

In terms of model refinement and variation, one possible enhancement is incorporating a geographical and temporal reference extraction model such as [12], which could help to detect scene-related entities and changes more effectively. Additionally, decoder-only models, such as GPT-3 [13] or GPT-4 [14] should be explored for this task. These models could be used in a zero-shot or few-shot learning setting, where they classify scene boundaries with little to no task-specific training data. Another possible direction would be to develop a model that, as with our CRF approach, treats scene segmentation as a sequence-labeling task at the whole sentence level, but uses sentence embeddings as sentence representations. I.e., instead of handling segmentation as a classification task at the sentence or sentence pair level, as we do in our BERT-based models, this approach would learn to assign sentence-level labels across sequences of sentence embeddings, potentially making better use of longer range contextual dependencies.

Another interesting and potentially important refinement could be incorporating the detection of non-scene segments and scene transition segments (STSs) into the task. Learning to identify these segments explicitly could potentially improve scene segmentation accuracy and would also result in a better representation the overall narrative structure.

Acknowledgements

The authors thank the Text2Story reviewers for their helpful comments. The first author acknowledges support from the University of Jeddah in the form of a PhD studentship.

References

[1] R. Gaizauskas, T. Alrashid, SceneML: A proposal for annotating scenes in narrative text, in: Proceedings of the 15th Workshop on Interoperable Semantic Annotation (ISA-15), Gothenburg, Sweden, 2019.

[2] H. Kozima, T. Furugori, Segmenting narrative text into coherent scenes, Literary and Linguistic Computing 9 (1994) 13–19.

[3] D. Kauchak, F. Chen, Feature-based segmentation of narrative documents, in: Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing - FeatureEng '05, June, Association for Computational Linguistics, Morristown, NJ, USA, 2005, p. 32. URL: http://www.aclweb.org/anthology/W/W05/W05-0405. doi:10.3115/1610230.1610237.

[4] J. E. Cutting, Event segmentation and seven types of narrative discontinuity in popular movies, Acta Psychologica 149 (2014) 69–77. URL: http://linkinghub.elsevier.com/retrieve/pii/S000169181400078X. doi:10.1016/j.actpsy.2014.03.003.

[5] A. Zehe, L. Konle, L. K. Dümpelmann, E. Gius, A. Hotho, F. Jannidis, L. Kaufmann, M. Krug, F. Puppe, N. Reiter, et al., Detecting scenes in fiction: A new segmentation task, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 3167–3177.

[6] T. Alrashid, R. Gaizauskas, ScANT: A small corpus of scene-annotated narrative texts, in: Proceedings of the Text2Story'23 Workshop, 2023, pp. 143–149.

[7] R. Gaizauskas, J. Wang, A. Ramisa, Defining visually descriptive language, in: Proceedings of the Fourth Workshop on Vision and Language, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 10–17.

[8] A. S. Maiya, ktrain: A low-code library for augmented machine learning, arXiv preprint arXiv:2004.10703 (2020). arXiv:2004.10703.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[10] T. W. MacFarland, J. M. Yates, T. W. MacFarland, J. M. Yates, Mann–whitney u test, Introduction to nonparametric statistics for the biological sciences using R (2016) 103–132.

[11] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 19–27.

[12] I. Ezeani, P. Rayson, I. N. Gregory, Extracting imprecise geographical and temporal references from journey narratives., in: Text2Story@ ECIR, 2023, pp. 113–118.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[14] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).