AI Summative Coursework

1. Task 1 - Word Clustering Using Unsupervised Learning

1.1. Introduction

This task explored whether semantically meaningful word clusters could be recovered from sentence-level co-occurrence patterns using unsupervised learning. Clustering algorithms were applied to a small corpus of Gothic fiction to investigate if thematic groupings emerged without labelled supervision.

The goal was to assess how well co-occurrence-based representations capture latent structure in a narrowly scoped domain and to evaluate the strengths and limitations of hierarchical and flat clustering methods for this purpose.

1.2. Methodology

1.2.1. Corpus Contruction and Preprocessing

Three Gothic novels, *Dracula*, *Frankenstein*, and *The Picture of Dorian Gray*, were sourced from Project Gutenberg and combined into a single text corpus. These works were selected for their shared genre traits and consistent narrative style.

The text was converted to lowercase and segmented into sentences. Each sentence was tokenized into words, with punctuation and non-alphabetic tokens removed. Single-character alphabetic tokens were retained if meaningful. The result was a nested list of cleaned, tokenized sentences used to compute co-occurrence patterns in later stages.

1.2.2. Content Word Selection

To support semantically meaningful clustering, a filtered list of content words was created by selecting the most frequent nouns in the corpus. This was based on the assumption that nouns typically represent concrete or abstract entities and are more likely to form coherent clusters than function words or modifiers.

The tokenized sentence structure was flattened into a single list of word tokens. Common stopwords were removed using a standard English stopword list, and tokens with fewer than two characters were excluded to reduce noise. A frequency distribution was computed, and the top 500 candidates were retained.

Part-of-speech tagging was then applied, and only tokens tagged as singular or plural nouns were kept. From these, the 100 most frequent were selected as the final content word list, denoted `L`. Common terms such as `time`, `man`, `life`, `eyes`, and `room` reflected the emotional and symbolic themes of the corpus.

1.2.3. Distance Matrix Construction

To measure semantic similarity, a pairwise distance matrix was built using sentence-level co-occurrence. For each sentence, all words from the list `L` (100 high-frequency nouns) were identified, and unordered word pairs were extracted. A co-occurrence count was incremented for each pair appearing together, resulting in a symmetric 100×100 matrix where each entry recorded the number of shared sentences.

To convert these counts into distances, an inverse transformation was applied as shown in Equation 1:

$$\text{distance} = \frac{1}{\text{co\_occurrence} + 1} \tag{1}$$

This ensured that strongly associated words had lower distances, while the addition of one prevented division by zero. The matrix diagonal was set to zero to reflect self-distance.

Sentence-level co-occurrence was chosen over average positional proximity due to its robustness to sentence length and its ability to reflect distributional similarity (Mikolov et al., 2013).

1.2.4. Hierarchical Clustering (HAC) Setup

Hierarchical Agglomerative Clustering (HAC) was applied to the pairwise distance matrix derived from sentence-level co-occurrence. HAC was selected for its ability to operate directly on custom distance matrices without requiring vector-space embeddings, and for its capacity to reveal nested semantic structure through dendrograms (Rokach and Maimon, 2005).

Clustering was performed using five linkage methods: Ward, average, complete, single, and centroid. Each defines inter-cluster distance differently (for example, Ward minimises variance, while single linkage uses nearest neighbours). All methods were applied to the same condensed distance matrix for comparability.

The number of clusters (k) was selected using silhouette-based model selection. As shown in Figure 1, scores were computed across a range of k values, and $k = 5$ was chosen to balance interpretability and structural separation.

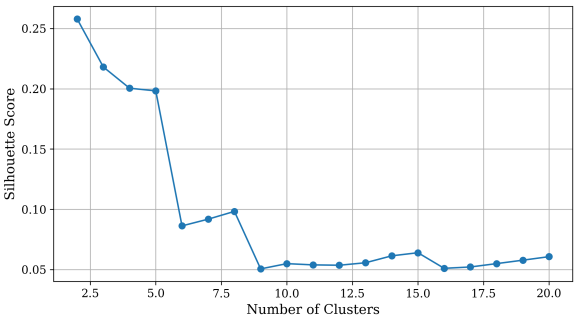

Figure 1: Silhouette score curve used to select number of clusters.

This HAC configuration served as the baseline for later experiments involving Dijkstra-based distance refinement and a flat clustering comparison using K-Means.

1.2.5. Dijkstra-Based Distance Refinement

To capture transitive semantic relationships that may not be evident from direct co-occurrence, Dijkstra's algorithm was used to compute shortest-path distances between words in a co-occurrence graph. Each node in the graph represented a word from the list `L`, and undirected edges were added between words that co-occurred in at least one sentence. Edge weights were defined using the same inverse transformation described in Equation 1, so strongly associated word pairs were connected by shorter edges.

The algorithm then computed the shortest cumulative edge weight between all word pairs, allowing indirect semantic links to be incorporated where direct connections were sparse. To ensure symmetry, the minimum of the forward and reverse path lengths was retained for each pair. Missing values were imputed using the best available alternative, and diagonal values were set to zero. The resulting distance matrix was then used as input to both HAC and K-Means, enabling direct comparison with the original co-occurrence-based clustering.

1.2.6. Evaluation Procedure

Clustering performance was evaluated using silhouette score and manual inspection. Silhouette scores, ranging from -1 to 1, were computed from co-occurrence-based feature vectors to assess intra-cluster cohesion and inter-cluster separation (Rousseeuw, 1987).

Qualitative analysis focused on whether groupings reflected meaningful categories such as emotions, body parts, characters, or abstract concepts. Shifts in thematic clarity were noted across methods, especially with Dijkstra-based refinement.

Together, these approaches provided a balanced view of clustering quality, combining quantitative performance with semantic interpretability.

1.3. Results and Discussion

1.3.1. Hierarchical Clustering Results (HAC)

Hierarchical Agglomerative Clustering (HAC) was initially applied to the pairwise distance matrix derived from sentence-level co-occurrence. Multiple standard linkage methods were evaluated to assess their impact on cluster structure and thematic cohesion.

Clustering quality across linkage methods was evaluated using silhouette scores, summarised in Table 1. Ward linkage achieved the highest score (0.198), indicating superior intra-cluster cohesion and inter-cluster separation. Average and centroid linkage produced less stable structures, while single linkage suffered

from chaining, and complete linkage collapsed most words into a single cluster.

Table 1: Silhouette scores for different HAC linkage methods.

Linkage Method	Silhouette Score
Ward	0.198
Average	-0.112
Centroid	-0.150
Single	-0.215

Based on this evaluation, Ward linkage was selected for detailed analysis, having produced the most coherent and interpretable groupings. The corresponding dendrogram is shown in Figure 2, illustrating a clear hierarchical organisation with well-separated clusters.

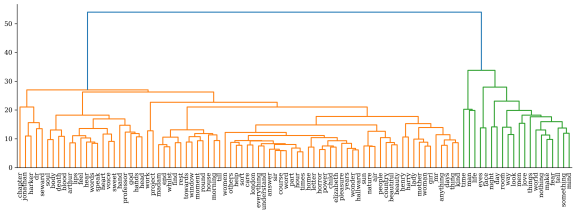

Figure 2: Dendrogram showing 5-cluster solution using Ward linkage.

Although the highest silhouette score was observed at $k = 2$ (see Figure 1), this configuration was too coarse to capture the semantic diversity of the corpus. A value of $k = 5$ was adopted based on the trade-off between cohesion and interpretability, as discussed in the methodology. This configuration was used as the baseline for comparing the effects of alternative clustering strategies and distance refinements.

The final Ward-based configuration produced five semantically distinct clusters, summarised in Table 2. These included named entities and abstract or narrative-related terms. Clusters also emerged around existential, physical, and emotional concepts.

1.3.2. Flat Clustering Comparison (K-Means)

As a baseline comparison, K-Means clustering was also applied using the co-occurrence matrix as feature input. After standardisation, the model was run with $k = 5$ to match the HAC configuration.

Table 2: Summary of Ward-based clusters and sample words.

Cluster	Thematic Focus and Sample Words
1	Named entities — `jonathan`, `dr`, `harker`, `chapter`, `seward`
2	Abstract and narrative mix — `heart`, `god`, `window`, `home`, `professor`, ... (+65 others)
3	Existential concepts — `time`, `man`, `life`
4	Physical descriptors — `eyes`, `face`
5	Emotions and experiences — `look`, `love`, `nothing`, `mind`, `fear`

The resulting silhouette score (0.149) was lower than HAC with Ward linkage (0.198), and although some thematic coherence was observed, the clusters lacked the hierarchical structure and interpretability of the HAC results.

1.3.3. Dijkstra-Based Refinement Results

To evaluate whether indirect semantic relationships could improve clustering, Dijkstra's algorithm was applied to the co-occurrence graph. Words were treated as nodes, with edge weights defined as the inverse of co-occurrence frequency plus one. All-pairs shortest-path distances were computed and used as input to HAC with Ward linkage.

The resulting dendrogram (Figure 3) was more flattened and diffuse, with one dominant cluster and a few smaller ones. Although the cluster sizes appeared more balanced, thematic coherence was reduced due to indirect links formed via high-frequency intermediary terms.

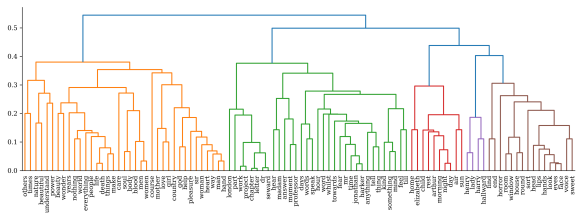

Figure 3: HAC dendrogram after applying Dijkstra-based refinement.

While shortest-path refinement may benefit structured domains, it weakened semantic

separation in this literary context. Sentence-level co-occurrence preserved clearer thematic boundaries, reaffirming HAC with Ward linkage as the more interpretable configuration.

A similar pattern was observed with K-Means. Performance comparisons before and after applying Dijkstra are summarised in Table 3.

Table 3: Silhouette scores before and after applying Dijkstra's algorithm.

Method	Pre-Dijkstra	Post-Dijkstra
HAC (Ward)	0.198	-0.027
K-Means	0.149	0.134

1.4. Conclusion

This project evaluated unsupervised clustering for uncovering latent semantic structure in a corpus of Gothic fiction. A sentence-level co-occurrence matrix was constructed for the most frequent nouns, and meaningful clusters were identified using Hierarchical Agglomerative Clustering (HAC). Ward linkage applied to raw co-occurrence distances yielded the most coherent and interpretable results, achieving the highest silhouette score.

K-Means also recovered plausible groupings, though with slightly lower cohesion and no hierarchical structure. In contrast, applying Dijkstra's algorithm led to performance decline, as shortest-path smoothing weakened thematic separation.

These findings support sentence-level co-occurrence as an effective distance metric and confirm HAC with Ward linkage as the most reliable method for extracting thematic clusters in narrative text.

2. Task 2 - Supervised Learning with Non-Linear Boundaries

2.1. Introduction

This task examined binary classification using synthetic two-dimensional data with a non-linear boundary. Logistic regression and a small neural network were compared across experiments varying boundary curvature, dataset size, class imbalance, and network architecture. These experiments aimed to assess how task complexity affects generalisation and to explore trade-offs between model simplicity and representational capacity.

2.2. Methodology

2.2.1. Data Generation

A synthetic dataset of 500 two-dimensional points was generated by sampling uniformly from the range [–3, 3] along both axes. Class labels were assigned based on whether each point lay above or below a quadratic decision boundary defined by:

$$y = a \cdot x^2 + x \tag{2}$$

Here, a controls the curvature of the boundary. Points above the curve were labelled Class 1, others Class 0.

This setup created a reproducible binary classification task with balanced coverage across the input space, following a common benchmarking approach in machine learning (Pedregosa et al., 2011).

Figure 4 illustrates the dataset for $a = 0.5$, with the decision boundary shown as a dashed curve.

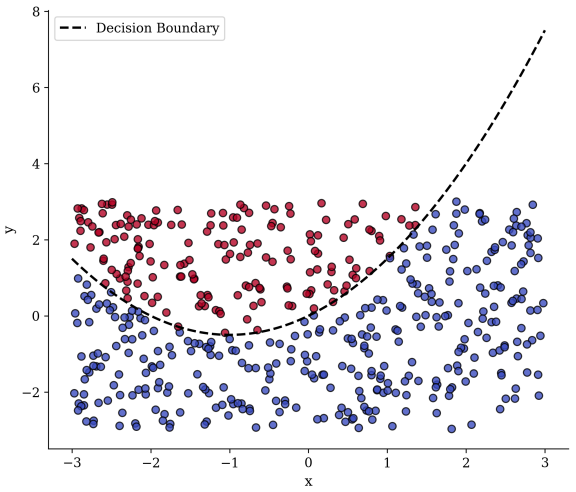

Figure 4: Synthetic dataset with decision boundary defined by $y = 0.5x^2 + x$.

2.2.2. Evaluation Setup

Model performance was evaluated using accuracy, precision, recall, and F1 score. As class imbalance was expected, F1 score was used as the primary metric due to its ability to balance precision and recall, which better reflect performance under skewed distributions (Saito and Rehmsmeier, 2015).

A series of controlled experiments was conducted to assess how different aspects of task complexity affect classifier performance. In

each experiment, one variable was changed while others were held constant. Both models were evaluated using the same data generation process and the metrics described above.

To test boundary curvature, the parameter a (Equation 2) was varied from 0.0 to 3.0, with error plots generated for $a = 0.0$, 1.0, and 3.0. For dataset size, training sets ranged from 50 to 2000 points, using $a = 1.0$. Class imbalance was introduced by vertically shifting the decision boundary in a 1000-point dataset to produce minority class proportions of 20%, 15%, 11%, and 8%. To assess model complexity, five neural network architectures ((5,), (10,), (20,), (20, 10), and (30, 20, 10)) were compared, to investigate potential diminishing returns with increased depth (Goodfellow et al., 2016).

2.3. Results and Discussion

2.3.1. Baseline Performance Comparison

Both classifiers were evaluated on a moderately non-linear dataset of 500 points with curvature $a = 0.5$. Table 4 summarises the results: logistic regression achieved an F1 score of 0.796, while the neural network reached 0.967, reflecting its superior capacity to model non-linear boundaries.

Figures 5 and 6 show model predictions on the same dataset. In all error plots, misclassified points appear in red and correctly classified points in white. Logistic regression produced errors near the curved boundary due to its linear constraint, which limits its ability to model non-linear structures, while the neural network misclassified only a few marginal points, successfully capturing the non-linear structure.

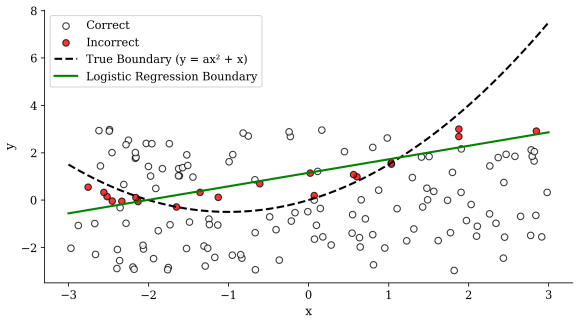

Figure 5: Logistic regression predictions on dataset with $a = 0.5$.

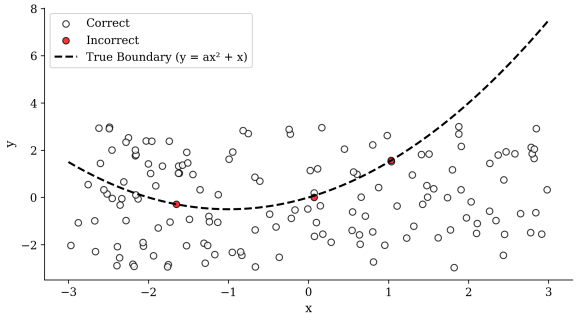

Figure 6: Neural network predictions on dataset with $a = 0.5$.

Table 4: Baseline performance metrics for logistic regression and neural network ($a = 0.5$).

Metric	Log Reg	Neural Net
Accuracy	0.873	0.980
Precision	0.787	0.978
Recall	0.804	0.957
F1 Score	0.796	0.967

2.3.2. Effect of Class Imbalance

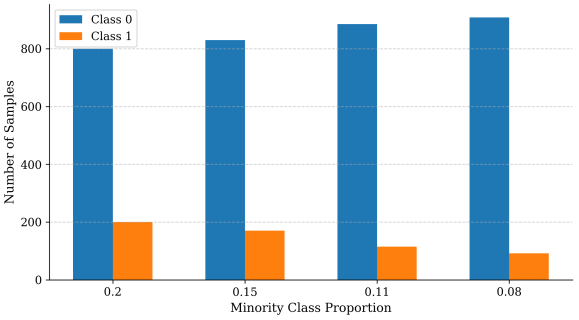

Figure 7: Class distribution under varying levels of imbalance.

Figure 7 shows class distributions for four imbalance levels, with minority proportions ranging from 20% to 8%. The neural network maintained high F1 scores across all settings, while logistic regression deteriorated sharply. In the most extreme case, its F1 score approached zero despite stable accuracy.

These results highlight the limits of accuracy under skew and the advantage of F1 score in reflecting true performance. The neural network's strength stems from its capacity to learn from sparse minority examples.

2.3.3. Effect of Boundary Curvature

As shown in Figure 8, both classifiers performed well at $a = 0.0$, but diverged rapidly as curvature increased. Logistic regression

deteriorated significantly, with F1 dropping below 0.5 at $a = 1.0$ and approaching zero at $a = 3.0$. In contrast, the neural network maintained high F1 scores and correctly classified most samples. Although logistic regression appears to misclassify only a small number of points, most belong to the minority class, which disproportionately impacts the F1 score.

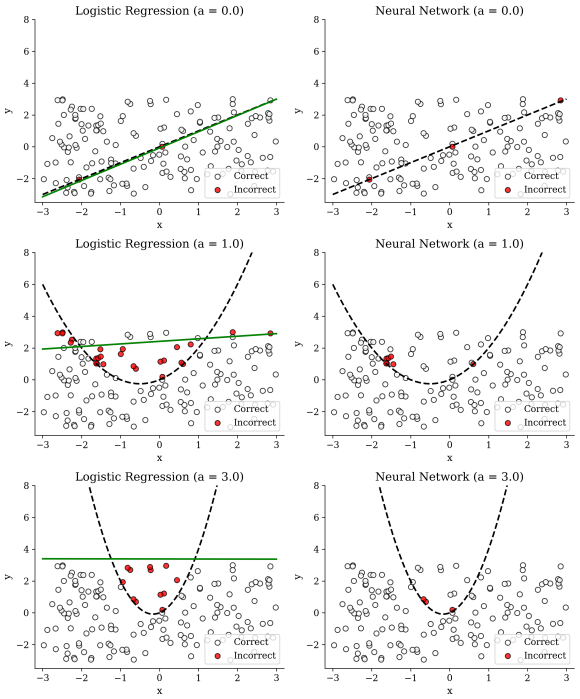

Figure 8: Error plots for curvature settings $a = 0.0$, 1.0, and 3.0.

2.3.4. Effect of Neural Network Architecture

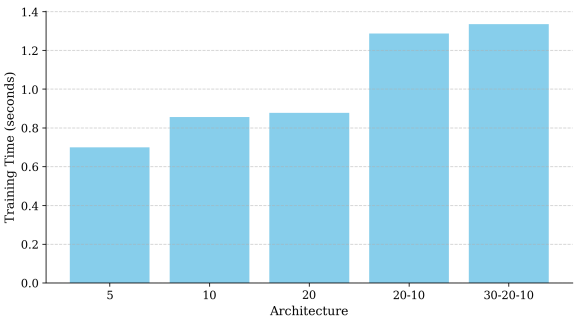

Figure 9: Training time comparison across neural network architectures.

F1 score improved with model depth, reaching perfect performance at the (20,) configuration. As shown in Figure 9, deeper architectures increased training time with no added benefit beyond moderate complexity.

These results reflect diminishing returns: once sufficient capacity is reached, additional layers increase cost without improving performance.

2.3.5. Effect of Dataset Size

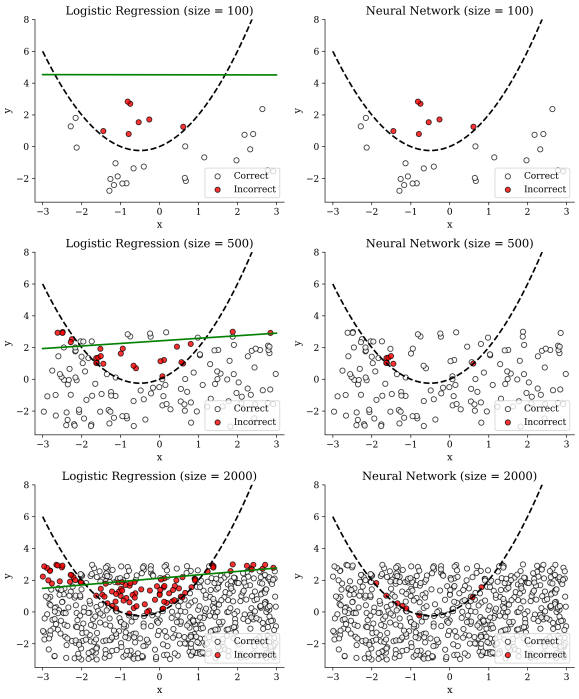

Figure 10: Error plots for dataset sizes 100, 500, and 2000.

As Figure 10 shows, the neural network improved rapidly with more training data, achieving near-perfect separation by 200 samples. Logistic regression improved only marginally, with persistent boundary errors even at 2000 samples, indicating bias rather than data scarcity. This was reflected in the F1 scores as well, which plateaued for logistic regression but approached 1.0 for the neural network.

2.4. Conclusion

This study compared logistic regression and neural networks on tasks with non-linear boundaries, imbalance, limited data, and varying model complexity. Logistic regression struggled under complexity, while neural networks consistently performed well. F1 score proved more informative than accuracy, especially under skew. Beyond a certain depth, network complexity added training cost without gains. Matching model capacity to task demands remains essential.

3. Task 3 - Do LLMs have rights?

My curiosity about artificial intelligence began long before today's large language models existed. I remember watching *I, Robot* as a child and becoming fascinated by the idea of machines becoming self-aware. I wondered if something like that could ever become real. That moment stayed with me and led to a question I have thought about ever since: could an artificial being ever deserve rights?

Years later, I find myself returning to that same question, now shaped by both personal interest and academic study. I regularly explore AI through articles and videos to understand how these systems work and where they are heading. The ethical questions discussed in our TIBS module, especially the Responsible Research and Innovation framework, helped sharpen my thinking. This combination of early curiosity and academic reflection has led me to a clear view that at their current stage, LLMs do not meet the conditions required for rights or personhood.

The concept of rights is closely tied to moral status, which is usually linked to beings with consciousness, sentience, or autonomous thought. As Martha Lewis explained in our Ethics lectures (Lewis, 2024), rights are protections for entities that can experience harm or meaningful consequences. Bostrom and Yudkowsky (Bostrom and Yudkowsky, 2014) argue that two key criteria for moral status are sentience, the ability to feel pain or pleasure, and sapience, which refers to higher reasoning and self-awareness. These qualities are currently found only in humans and animals.

This leads to a deeper question. Do LLMs show traits that resemble personhood, even if they lack moral status? Traits like memory, language, reasoning, and goal-driven behaviour are often seen as part of what makes something a person. LLMs seem to mimic some of these abilities, especially in conversation and problem-solving. However, they do this without self-awareness, a continuous sense of self, or any inner experience. Their outputs are generated by algorithms trained on large datasets to predict likely responses. Bender et al. (Bender et al., 2021) describe LLMs as "stochastic parrots", meaning they generate fluent language by probabilistically assembling fragments of their training data without any real understanding. This reinforces the view that LLMs, while appearing articulate, are fundamentally pattern recognition systems rather than conscious agents. Since they only simulate these traits and lack genuine understanding, they do not meet the requirements for moral consideration under current ethical standards.

As AI becomes more involved in real-world decisions such as driving cars, operating machinery or assisting customers, it becomes harder to define who is responsible when something goes wrong. In these situations, moral and legal responsibility can become unclear, especially when the outcome is not directly caused by a programmer. Some scholars, including Gunkel (Gunkel, 2018), argue that legal systems may need to treat certain AI systems as having limited personhood. This would not be based on sentience but on the need to manage accountability. Personhood in this context is a legal tool, not a moral judgement. While I do not believe LLMs deserve rights in the ethical sense, I recognise that legal systems may need to adapt as these technologies take on more responsibility.

Looking ahead, it is important to stay open to the possibility that AI could one day develop in ways that challenge our current ethical views. If a system were to gain sentience, sapience, or the ability to suffer, then its moral status would need to be reconsidered. Bostrom and Yudkowsky (Bostrom and Yudkowsky, 2014) argue that what matters is not being human but having the capacity to think and feel. In that case, the idea of rights would need to go beyond biology. For now, LLMs remain powerful but non-conscious tools.

In conclusion, large language models are an impressive step in technology, but they do not meet the conditions for rights or personhood. My experiences have led me to believe that ethical responsibility must remain with the people who design and deploy these systems. Frameworks like Responsible Research and Innovation help guide how we build AI today. Until sentience or true understanding emerges, our focus should remain on using these tools responsibly, transparently, and with human accountability.

4. References

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, Virtual Event, Canada. ACM.

Nick Bostrom and Eliezer Yudkowsky. 2014. The ethics of artificial intelligence. In Keith Frankish and William M. Ramsey, editors, *The Cambridge Handbook of Artificial Intelligence*, pages 316–334. Cambridge University Press, Cambridge.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

David J. Gunkel. 2018. *Robot Rights*. MIT Press, Cambridge, MA.

Martha Lewis. 2024. Ethics. Lecture notes, Department of Engineering Mathematics, University of Bristol. Accessed during Ethics lecture.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Lior Rokach and Oded Maimon. 2005. Clustering methods. In *Data Mining and Knowledge Discovery Handbook*, pages 321–352. Springer.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432.