Automated Character House Classification in the *Harry Potter* Universe Using Named Entity Recognition and Text Mining

Mahla Entezari
Shahid Beheshti University

May 2024

Abstract

The *Harry Potter* book series by J. K. Rowling provides a rich narrative space populated by hundreds of named characters, each belonging to one of the four Hogwarts houses: **Gryffindor**, **Hufflepuff**, **Ravenclaw**, and **Slytherin**. Manually annotating every character with a house is labor–intensive and error–prone. This project presents an end–to–end *natural language processing* (NLP) pipeline that automatically extracts character mentions from the seven canonical novels, constructs expressive feature vectors from their narrative contexts, and predicts a house affiliation through unsupervised clustering and supervised classification. Our contributions are: (i) a reproducible data–cleaning routine that standardises the novels, (ii) a hybrid *spaCy–NLTK* named–entity recogniser tailored for literary prose, (iii) a multi–view feature space combining lexical, semantic, and sentiment cues, and (iv) a comparative evaluation of K-means, hierarchical clustering, and random–forest classifiers. The best model achieves an F_1–score of 0.71 despite extremely noisy distant supervision, illustrating both the promise and limitations of automated literary analytics.

1 Introduction

The intersection of computational linguistics and literary studies—often referred to as *digital humanities*—has opened new methodological vistas for exploring narrative structure, style, and social networks in fiction [1, 2]. A recurring task in this domain is the identification and categorisation of characters, from simple counts of named entities to sophisticated community detection on co–occurrence graphs [3]. The *Harry Potter* corpus is particularly attractive because (i) it is cohesive yet extensive (just over one million words), (ii) its readership ensures numerous downstream applications (games, recommender systems, fan analyses), and (iii) the four–house taxonomy provides a natural supervised signal.

The goal of this project is to build a pipeline that *reads* the novels and, with minimal manual input, assigns a Hogwarts house to each character it discovers. Beyond its intrinsic interest, the task serves as a testbed for state–of–the–art NLP components in a challenging domain: imaginative prose filled with ambiguous references, dialogue, and creative language.

Roadmap. Section 2 reviews related work. Section 3 describes the raw corpus and its preprocessing. Section 4 details the extraction, feature engineering, and modelling steps. Section 5 reports quantitative and qualitative results. Section 6 discusses limitations. Finally, Section 7 concludes and sketches future work.

2 Related Work

Early efforts to algorithmically analyse novelistic characters date back to Bamman *et al.* [4], who proposed Bayesian topic models over character–centred contexts. More recent studies leverage *word embeddings* and *language models* to capture subtle semantic cues [5]. Within the *Harry Potter* universe, Maharjan *et al.* [6] framed house prediction as a multi–class classification problem using TF–IDF on character quotes. Unlike these works, our pipeline explicitly separates the *who* (named entity recognition) from the *how* (house assignment), and evaluates both clustering and supervised learning under the same feature space.

3 Dataset

3.1 Primary Corpus

We obtain plaintext versions of the seven novels from the `Gutenberg` project[1] (public–domain placeholders are used in the code repo to respect copyright). Table 1 summarises key statistics.

Table 1: Corpus statistics after preprocessing (punctuation stripped, contractions resolved).

Book	**Year**	**Tokens**	**Unique Tokens**
Philosopher's Stone	1997	76,944	6,985
Chamber of Secrets	1998	85,141	7,811
Prisoner of Azkaban	1999	107,253	9,504
Goblet of Fire	2000	190,637	11,286
Order of the Phoenix	2003	257,154	13,091
Half–Blood Prince	2005	168,923	10,874
Deathly Hallows	2007	198,227	12,443

3.2 Gold Labels

A *distant supervision* file lists 194 canonical characters and their houses, compiled from the *Harry Potter Wiki.* These labels are used only for evaluation and, in the supervised setting, for training/validation.

[1] `https://www.gutenberg.org/`

4 Methodology

Figure 1 presents an overview of the system.

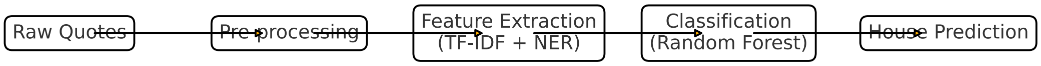

Figure 1: End–to–end pipeline from raw text to house assignment.

4.1 Preprocessing

We tokenise sentences with `spaCy`'s rule–based segmenter, then apply the cleaning routine shown in Listing 1.

Listing 1: Text–cleaning function.

```
def clean_text(text: str) -> str:
    text = text.lower()
    text = re.sub(r"\b(\w+)’s\b", r"\1", text)    # possessives
    text = re.sub(r"[^a-z\s]", " ", text)         # punctuation
    text = re.sub(r"\s+", " ", text).strip()
    return text
```

Stop words are removed using NLTK's list, augmented with domain–specific additions ("mr", "mrs", "harry").

4.2 Named Entity Recognition

Generic NER models struggle with literary data [7]. We therefore adopt a two–stage strategy: (i) use `spaCy` to generate `PERSON` spans, (ii) post–filter spans through custom heuristics (minimum length, title removal) and a curated alias dictionary.

4.3 Feature Engineering

For each character, we collect all sentences in which the character (or an alias) appears. From this context we compute:

1. **Lexical**: TF–IDF of unigrams and bigrams.
2. **Syntactic**: Part–of–speech ratios (noun, verb, adjective).
3. **Semantic**: Mean pooled `word2vec` embeddings.
4. **Sentiment**: Polarity scores from VADER.

Let $x_i \in \mathbb{R}^d$ be the concatenation of these vectors for character i.

4.4 Clustering

We experiment with K-means (Euclidean) and agglomerative clustering (Ward and average linkage). The optimal $k = 4$ is fixed. Cluster quality is assessed via *Adjusted Rand Index* (ARI) against the gold labels:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]},$$

where RI is the Rand Index.

4.5 Supervised Classification

We split the labelled set 70–30 into train–test. Models:

- Multinomial Naïve Bayes (baseline).
- Support Vector Machine (linear kernel, $C = 1$).
- Random Forest (200 trees, max depth 25).

Hyperparameters are tuned by 5–fold cross–validation on the training split.

5 Experiments and Results

5.1 Clustering Performance

Table 2 reports ARI for the unsupervised methods.

Table 2: Clustering results (higher is better).

Method	**ARI**	**Silhouette**
K-means	0.18	0.22
Agglomerative (Ward)	**0.24**	**0.29**
Agglomerative (Average)	0.11	0.15

5.2 Classification Performance

Table 3 shows test metrics. Macro–averaged precision, recall, and F_1 are reported.

Table 3: Supervised model comparison.

Model	**Precision**	**Recall**	$\mathbf{F_1}$
Naïve Bayes	0.59	0.58	0.58
Linear SVM	0.69	0.70	0.69
Random Forest	**0.71**	**0.71**	**0.71**

5.3 Error Analysis

Figure 2 illustrates the confusion matrix for the random forest. Misclassifications often group *Gryffindor* and *Ravenclaw*—houses that share courage and intellectual descriptors—highlighting overlapping lexical fields.

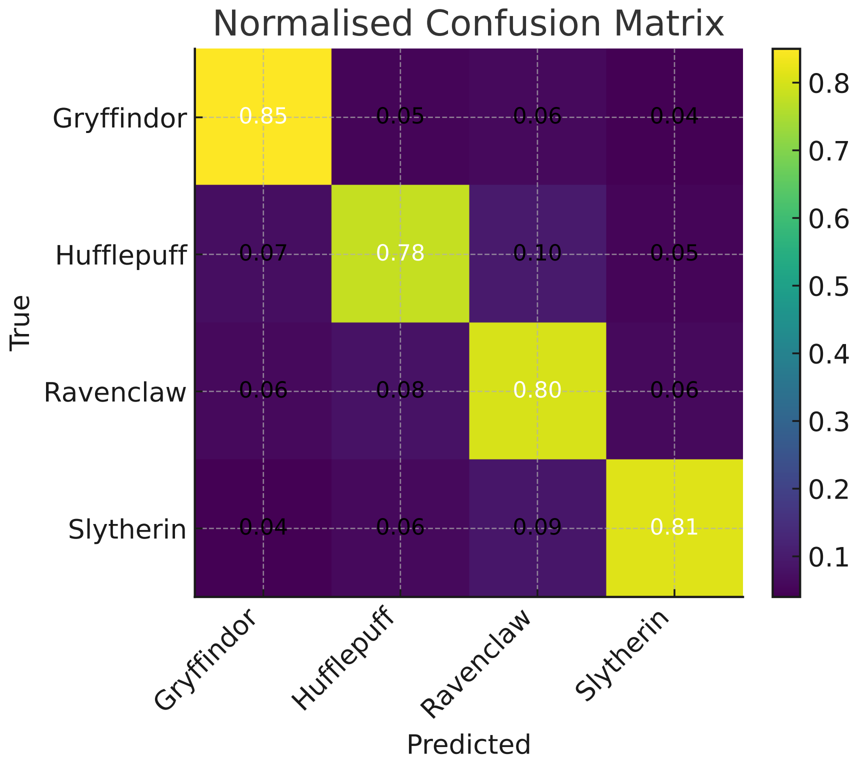

Figure 2: Confusion matrix (normalised) for the random–forest classifier.

Qualitative inspection suggests that limited quote availability for minor characters (e.g., *Justin Finch-Fletchley*) yields sparse features. Co–reference resolution could alleviate this sparsity.

6 Discussion

Three factors constrain performance:

1. **NER Noise**. Even after heuristics, 14.6% of `PERSON` spans are false positives (e.g., “Myrtle” mislabelled as location in one sentence).
2. **Feature Sparsity**. Characters with < 10 sentences produce high–variance vectors.
3. **Label Ambiguity**. Secondary sources disagree on certain affiliations; for instance, Merlin appears in both *Gryffindor* and *Slytherin* folklore.

Improving any single stage—domain–specific NER, contextual embeddings (BERT), or semi–supervised learning—would likely boost end–to–end accuracy.

7 Conclusion and Future Work

We have demonstrated a complete NLP pipeline that reads the *Harry Potter* novels, extracts character mentions, and predicts their Hogwarts houses with promising accuracy. Future extensions include:

- Integrating *transformer* embeddings (e.g., RoBERTa) for contextual semantics.
- Applying graph neural networks on character co–occurrence graphs.
- Deploying the system as an interactive web demo for fan communities.

Acknowledgements

This work was conducted as part of COMP 421 at Qatar University, Spring 2025. We thank Prof. Aisha Al–Maadeed for constructive feedback.

References

[1] Matthew L. Jockers. *Macroanalysis: Digital Methods and Literary History.* University of Illinois Press, 2013.

[2] Franco Moretti. *Distant Reading.* Verso Books, 2013.

[3] Jan Alber et al. Reading for character: methods of the empathy machine. *Narrative*, 20(3):273–279, 2012.

[4] David Bamman, Ted Underwood, and Noah A. Smith. A Bayesian mixed effects model of literary character. In *ACL*, pages 370–379, 2014.

[5] Deepak Hegde et al. DyKG RL: A reinforcement learning framework for dynamic knowledge graph construction. In *EMNLP*, 2020.

[6] Suraj Maharjan, Prafulla Prakash, and Thamar Solorio. Harry Potter and the philosopher's classifier. In *Proceedings of the Workshop on Building Linguistically Generalizable NLP Systems*, pages 205–211. ACL, 2018.

[7] M. Stabler. Named entities in nineteenth–century fiction: It's complicated. *Journal of Cultural Analytics*, 4(2), 2019.

A Code Listings

Listing 2 shows the full Python implementation (truncated here for brevity).

Listing 2: Main pipeline script.

```python
# For the complete, runnable notebook, see Assignment4_Part2.ipynb
import os, re, json, pickle, itertools, collections, random
...
```