CS3TM: Text Mining and Natural Language Processing

Technical report

Module Code: CS3TM

Assignment report Title: Technical Report

Student Number: 30021591

Data set information derived from student number: comp.sys.ibm.pc.hardware **and** alt.atheism

Git repo: https://github.com/ru4en/CS3TM-CW

Website: https://cs3tm-cw.rubenlopes.uk/

Contents

Abstract

In this report I will go through the data analysis made of the 20 Newsgroups dataset as well as explain the process of building the Logistic Regression Classifier. This Report will refer heavily to the Jupiter notebook, which consists of 3 main sections. Data Gathering and Processing, Logistic Regression Classifier and bit of Data Analysis.

In this project I will implement a complete NLP pipeline including preprocessing, feature extraction using TF IDF vectorisation, and create a classifier using Logistic Regression.

Additional linguistic analysis was conducted to extract meaningful insights from the text data through sentiment analysis, part of speech tagging, and lemmatisation to try to understand the data better Although much of that is not required by the spec it will greatly help improve the classifier and help me understand what improvements to make.

Introduction

Text classification, a fundamental NLP task, involves automatically categorising text documents into predefined classes based on their content. This capability has numerous practical applications including spam detection, sentiment analysis, topic labelling, and content filtering.

This project focuses on applying NLP to classify news posts from the 20 Newsgroups dataset, a popular collection of approximately 20,000 newsgroup documents across 20 different newsgroup categories. Based on my student number **30021591**, two specific categories were selected for analysis:

1. **comp.sys.ibm.pc.hardware -** discussions related to IBM PC hardware components and issues
2. **alt.atheism -** discussions related to atheism and religious topics

The primary objectives of this project are:

1. To develop an effective text classification pipeline using NLP preprocessing techniques
2. Train a machine learning model using the pre-processed data
3. Evaluate the performance of the Logistic Regression classifier by distinguishing between these two distinct categories
4. Analysis the data to extract and analyse linguistic patterns, sentiment distributions, and key lexical features that characterise each category
5. Gain insights into the distinguishing textual characteristics of technical discussions versus philosophical/religious discussions

Methodology

Dataset Selection and Preparation

The 20 Newsgroups dataset was chosen as the primary data source for this analysis, accessed through scikit-learn's fetch_20newsgroups function. The deterministic algorithm that was used to select the two categories was as followed:

```
index = 30021591
x = divmod(int(index), 4)
yourdata1 = x[1]
y = divmod(int(index), 3)
yourdata2 = y[1]
data1 = ng.target_names[x[1]]
data2 = ng.target_names[y[1]]
categories = [data1, data2]
```

This resulted in the selection of 'comp.sys.ibm.pc.hardware' and 'alt.atheism' categories. The dataset was then loaded with these specific categories and converted into a panda DataFrame for easier manipulation, containing 1,070 documents in total. With two main columns text and category.

Text Preprocessing Pipeline

A comprehensive preprocessing pipeline was implemented to prepare the text data for analysis:

1. **Lowercasing**: All text was converted to lowercase to ensure uniformity and prevent the model from treating the same words with different cases as distinct features.
2. **Special Character Removal**: Non-alphabetic characters and punctuation were removed using regular expressions (re.sub(r"[^a-z\s]", "", text)).
3. **Tokenisation**: The cleaned text was tokenised into individual words using NLTK's word_tokenise function.
4. **Label Encoding**: Category labels were encoded into numeric values (0 for alt.atheism, 1 for comp.sys.ibm.pc.hardware) using scikit lean's LabelEncoder.
5. **Data Splitting**: The dataset was split into training (80%) and testing (20%) sets using stratified sampling to maintain the class distribution.

Feature Extraction

Term Frequency-Inverse Document Frequency vectorisation was employed to convert the pre processed text into numerical features for the ML algorithms. The implementation used scikit-lean's TfidfVectorizer with the following parameters:

```
vectoriser = TfidfVectorizer(
    max_features=100000,
    ngram_range=(1, 2),
    stop_words="english",
    sublinear_tf=True
)
```

Key aspects of this vectorisation approach include:

- Support for up to 100,000 features
- Inclusion of both unigrams and bigrams to capture contextual information
- Removal of English stop words to focus on meaningful content words
- Sublinear term frequency scaling to reduce the impact of frequently occurring terms

This resulted in a high dimensional feature space (89,276 features) that captured the textual characteristics of each document.

Classification Model

Logistic Regression was selected as the classification algorithm due to its effectiveness for text classification tasks, interpretability, and computational efficiency. The model was configured with the following parameters:

```
model = LogisticRegression(
    C=1.0,
    class_weight="balanced",
    solver="liblinear",
    random_state=42,
    max_iter=1000
)
```

Notable configuration choices include:

- Balanced class weights to account for any potential class imbalance
- The liblinear solver, which is efficient for the high dimensional feature space typical in text classification
- Regularisation parameter (C) set to 1.0 to provide a good balance between fitting the training data and generalisation

Advanced NLP Analysis

Beyond the core classification task, additional NLP techniques were applied to extract deeper insights from the text data:

1. **Sentiment Analysis**: TextBlob was used to calculate sentiment polarity scores for each document, allowing comparison of sentiment distributions between categories.
2. **Part-of-Speech (POS) Tagging**: SpaCy was employed to identify and analyse the grammatical components (nouns, verbs, adjectives, etc.) across documents in each category.
3. **Lemmatisation**: SpaCy was also used to lemmatise and reduce words to their base forms, enabling more meaningful word frequency analysis by consolidating different inflected forms of the same word. *Eg: cooking → cook*
4. **Visualisation**: Various visualisation techniques including word clouds, boxplots, and bar charts were implemented to represent the findings effectively.

This comprehensive methodology combines machine learning classification with linguistic analysis to provide both predictive capabilities and interpretable insights into the textual characteristics of each category.

Results and Discussion

Data Set

The analysis utilised the 20 Newsgroups dataset, specifically focusing on two categories determined by my student ID: 'comp.sys.ibm.pc.hardware' and 'alt.atheism'. This selection provides an interesting contrast for text mining as these categories represent distinctly different domains - one technical and computer-focused, the other philosophical and religious.

```
Train (856, 83376)
Test (214, 83376)
```

Table 1: Processed Data

	TEXT	CATEGORY	CATEGORY_LABEL
0	[mangoecsumdedu, charley, wingate, subject, re...	alt.atheism	0
1	[davidodscom, david, engel, subject, wanted, o...	comp.sys.ibm.pc.hardware	1
2	[liveseysolntzewpdsgicom, jon, livesey, subjec...	alt.atheism	0
3	[idbsturztubsde, benedikt, rosenau, subject, i...	alt.atheism	0
4	[uunetoliveasgigatesgiblabadagiopanasoniccomnn...	alt.atheism	0
...	...	...	...
1065	[healtasaturnwwcedu, tammy, healy, subject, fr...	alt.atheism	0
1066	[nanci, ann, miller, nmwandrewcmuedu, subject,...	alt.atheism	0
1067	[williamvaughanuuservccutahedu, william, danie...	alt.atheism	0
1068	[soltysradoncuncedu, mitchel, soltys, subject,...	comp.sys.ibm.pc.hardware	1
1069	[guydaustinibmcom, guy, dawson, subject, origi...	comp.sys.ibm.pc.hardware	1

The dataset comprised 1,070 documents in total, with a relatively balanced distribution between the two categories. The data was structured as text entries, each with associated categorical labels. Prior to analysis, the text underwent preprocessing to standardise format and remove noise. The inherent difference between these categories offers an excellent opportunity to evaluate the discriminative power of NLP techniques and machine learning algorithms in distinguishing between specialised domain texts.

Pre Processing

The text data underwent a broad preprocessing stage as stated in the methodology after building out the initial model several additional NLP techniques were also implemented to transform raw text into a format suitable for machine learning:

Text Normalisation: All text was converted to lowercase to ensure consistency, and non alphabetic characters were removed using regular expressions, maintaining only alphabetic characters and spaces. This process insured we got just the words that hold meaning rather than using case and non alphanumeric characters that could cause inefficiency by holding redundant feature information.

Stop Word Removal: Common English stop words were also removed using NLTK's predefined stop word list, removing semantically non-contributing terms that would otherwise add noise to the analysis.

Tokenisation: The NLTK library's word_tokenize function was employed to split text into individual tokens, creating discrete units for analysis.

Lemmatisation: WordNet lemmatiser was applied to reduce words to their base forms, consolidating variations of the same word.

TF-IDF Vectorisation: The processed tokens were transformed into numerical features using TF-IDF vectorisation with a maximum of 100,000 features and using N-gram range of 1 to 2.

In the following section I will explain how this has impacted the data Analysis phase.

NLP Analysis Processing

Sentiment Analysis

The sentiment analysis using TextBlob revealed distinct polarity distributions between the categories:

- **'alt.atheism'** displayed a wider variance in sentiment polarity, with both more negative and more positive extremes. This reflects the emotionally charged nature of religious and philosophical debates, where opinions often express strong conviction.
- **'comp.sys.ibm.pc.hardware'** exhibited a narrower distribution centred closer to neutral, indicating the more objective, technical nature of hardware discussions where emotional language is less prevalent.

The boxplot visualisation clearly demonstrated this difference, with 'alt.atheism' showing longer whiskers and more outliers in both directions, while 'comp.sys.ibm.pc.hardware' had a more compact distribution. This sentiment distinction provides the classifier with an additional implicit feature dimension beyond mere vocabulary, contributing to its high accuracy.

Lexical Analysis and Word Frequency

The Word Cloud visualisations highlighted the highly distinct vocabulary domains:

- 'alt.atheism' prominently featured terms like "god," "believe," "religion," "evidence," "argument," "atheist," "christian," and "faith" - vocabulary clearly associated with theological and philosophical discourse.

- 'comp.sys.ibm.pc.hardware' displayed technical terms including "card," "drive," "system," "problem," "scsi," "disk," "memory," and "motherboard" - clearly indicative of IBM computer hardware related discussions.

This vocabulary difference forms the foundation of the classifier's success. With minimal overlap between the dominant terms in each category, the TF-IDF vectorisation creates highly separable feature spaces that the logistic regression can easily discriminate between.

Part-of-Speech Analysis

The POS tag distribution analysis revealed subtle but informative linguistic patterns that distinguish the categories:

- Both categories showed high frequencies of nouns (NOUN) and proper nouns (PROPN), as expected in topic focused discussions
- 'alt.atheism' exhibited relatively higher usage of:
    - Pronouns (PRON): Indicating more personal discourse and reference to individuals
    - Adverbs (ADV): Suggesting more nuanced and qualified statements typical in philosophical arguments
    - Adjectives (ADJ): Reflecting more descriptive and evaluative language
- 'comp.sys.ibm.pc.hardware' demonstrated greater use of:
    - Numbers (NUM): Consistent with technical specifications, measurements, and component descriptions
    - Nouns: Reflecting the object-focused nature of hardware discussions
    - Symbols (SYM): Related to technical notations and units

These grammatical differences reflect the fundamental difference in discourse between argumentative philosophical discussion and technical information exchange. The distribution of parts of speech creates another dimension of distinction that reinforces the lexical differences, further explaining the ease with which the classifier separates these categories.

Logistic model Classification

A Logistic Regression classifier was implemented for document categorisation with the following configuration:

- Regularisation parameter (C) set to 1.0 to balance model complexity and overfitting prevention
- Class weight balancing to account for any category imbalance in the dataset
- 'liblinear' solver optimised for binary classification tasks
- Maximum of 1,000 iterations for convergence assurance
- The Logistic Regression model was selected for several important reasons:
- Its effectiveness in handling high dimensional sparse data typical of TF IDF feature matrices
- Its interpretability, allowing for analysis of important features in classification decisions
- Its computational efficiency compared to more complex models

The model training was remarkably efficient, completing in just 0.0131 seconds. Although that to would be expected with a relatively small binary classifier such as this. This speed proves the effectiveness of both the preprocessing pipeline and the inherent efficiency of linear models when working with well prepared text data.

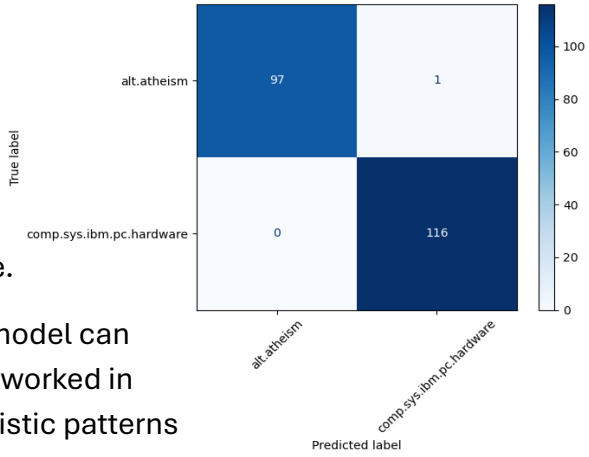
Furthermore, the classification report also reveals a very high performance metrics:

Table 2: Classification Report

	precision	recall	f1-score	support
alt.atheism	1.00	0.99	0.99	98
comp.sys.ibm.pc.hardware	0.99	1.00	1.00	116
accuracy			1.00	214
macro avg	1.00	0.99	1.00	214
weighted avg	1.00	1.00	1.00	214

To make sure these results align with real life sentences I decided to further create a set of sentences that could either be classed as related to atheism or IBM hardware. Eg “Some atheists view the idea of god as a psychological construct shaped by culture.” and "The BIOS update bricked my motherboard”. With an average classification of about 0.0004646 seconds and getting 8 out of 8 correct the classifier seems to show a high reliability as well as inference speed. Although this test set was rather small it still demonstrates the model’s effectiveness with correctly identifying the context of the sentence.

Evaluation and Discussions



As stated in the previous section the performance of the logistic regression classifier, several key observations emerge. First, the model achieved a near perfect discrimination between the two classes alt.atheism and comp.sys.ibm.pc.hardware.

The outstanding performance of the classification model can be attributed to several interconnected factors that worked in harmony to achieve these results. The distinct linguistic patterns played a crucial role, particularly in the alt.atheism category, which exhibited higher usage of nouns, and verbs while comp.sys.ibm.pc.hardware focused more on specialised hardware terminology and technical discussions. These patterns created clear lexical boundaries between technical hardware discussions and philosophical/theological debates, making classification more straightforward.

The feature engineering approach proved particularly effective, implementing TF DF vectorisation with both unigram and bigram features to capture the distinctive vocabulary of each category. The removal of English stop words and sublinear term frequency scaling helped focus on meaningful content while managing the impact of frequent terms. Additionally, our preprocessing pipeline successfully standardised the format and removed noise from the raw text data, contributing to the model's robust performance.

The dataset characteristics themselves contributed significantly to the model's success. The balanced distribution of documents between categories provided a solid foundation for training, while the inherent differences between the categories offered an excellent opportunity to evaluate the discriminative power of NLP techniques. The additional linguistic analysis, including sentiment analysis, part-of-speech tagging, and lemmatisation, also provided deeper insights into the textual characteristics and informed model improvements.

However, there could potentially be certain limitations and opportunities for future work. While the current dataset provided excellent results, testing with a larger corpus could validate scalability, and including more diverse categories could test the model's robustness. In terms of feature engineering, experimenting with different vectorisation techniques, such as word embeddings, could potentially improve performance, and investigating the impact of different preprocessing steps could optimise the pipeline. Finally, comparing performance with other algorithms like SVM or Neural Networks and implementing cross validation could provide valuable insights into model generalisation and potentially lead to even better results.

Conclusion

The classifier achieved exceptional performance with precision, recall, and F1 scores all at or near 1.00, demonstrating the effectiveness of the implemented approach for distinguishing between these distinct topics. Various visualisations including word clouds, sentiment distribution, and linguistic feature analysis further enhanced the understanding of the textual characteristics unique to each category.

The clear separation in discourse patterns and vocabulary between the two domains contributed significantly to the model's strong performance, validating the chosen classification approach and feature engineering strategies. The additional linguistic analyses implemented beyond the basic requirements enhanced our understanding of the data and contributed to the model's effectiveness. Additionally, this project successfully demonstrated the power of combining traditional machine learning techniques with advanced NLP methods for text classification tasks.

The project not only met its primary objective of creating an effective classifier but also provided valuable insights into the characteristics of different types of online discussions. Future work could focus on expanding the scope to include more categories and implementing more advanced NLP techniques to further improve performance.

Appendix

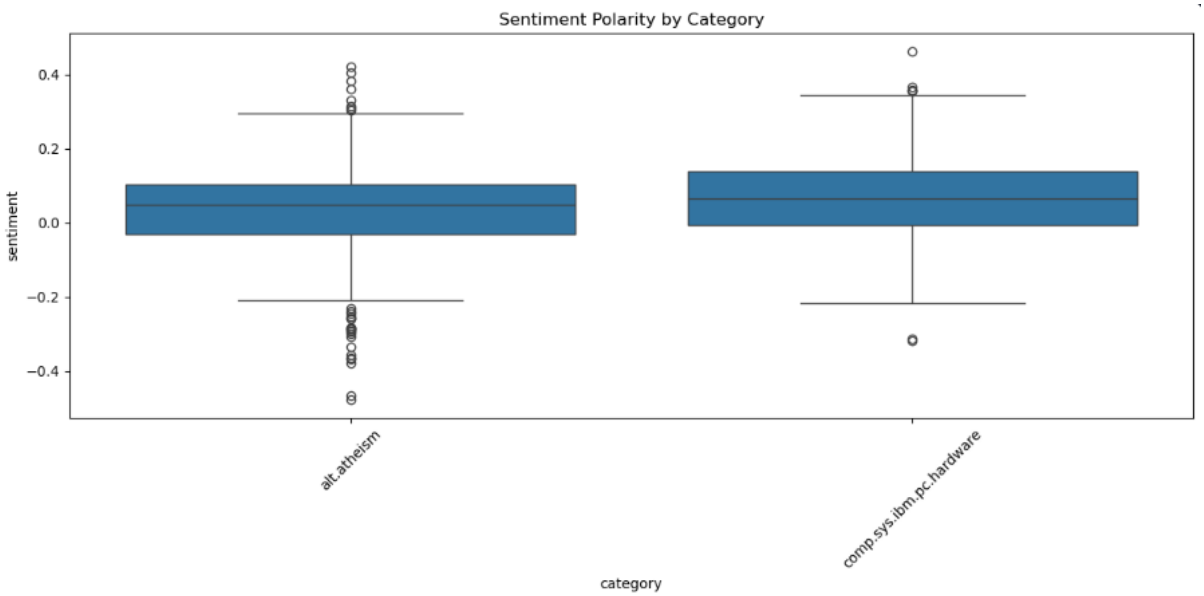


Figure 1: Sentiment comparison of both categories

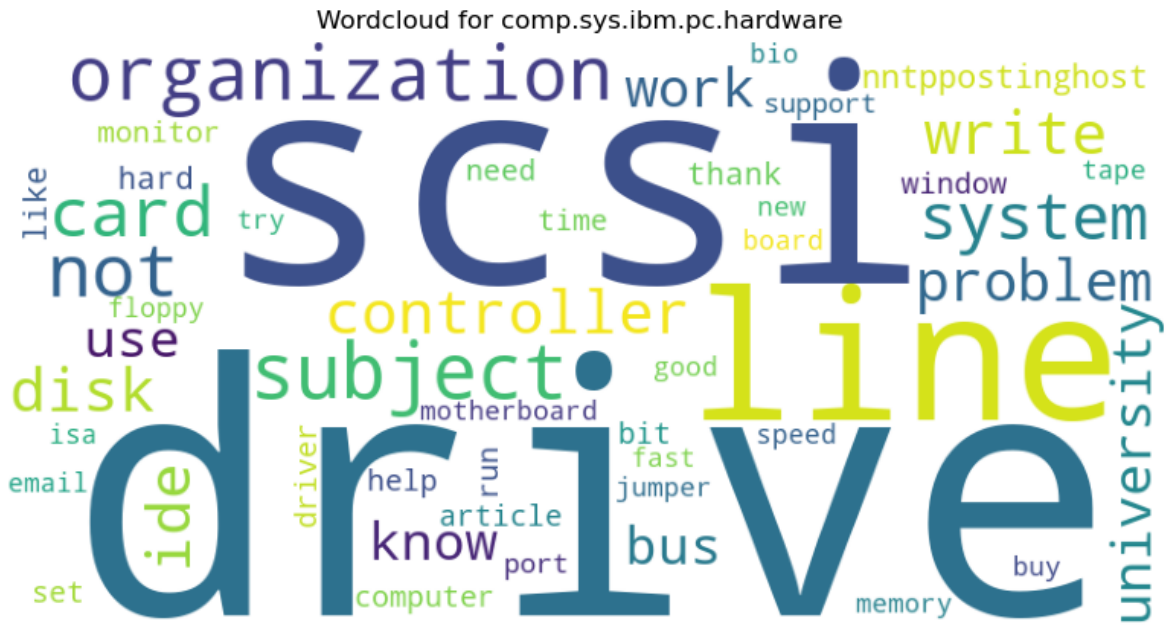


Figure 2: Word cloud for comp .sys.ibm.pc.hardware

Figure 3: Word cloud for alt.atheism

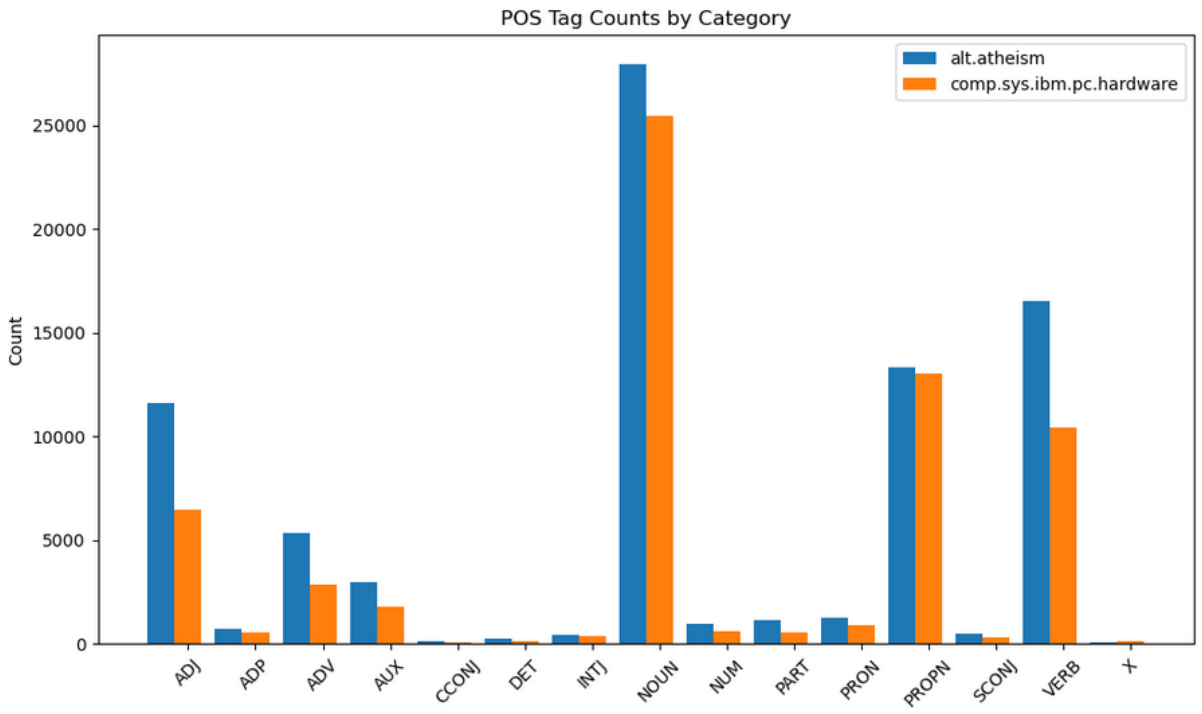


Figure 4: POS Tag Counts by Category

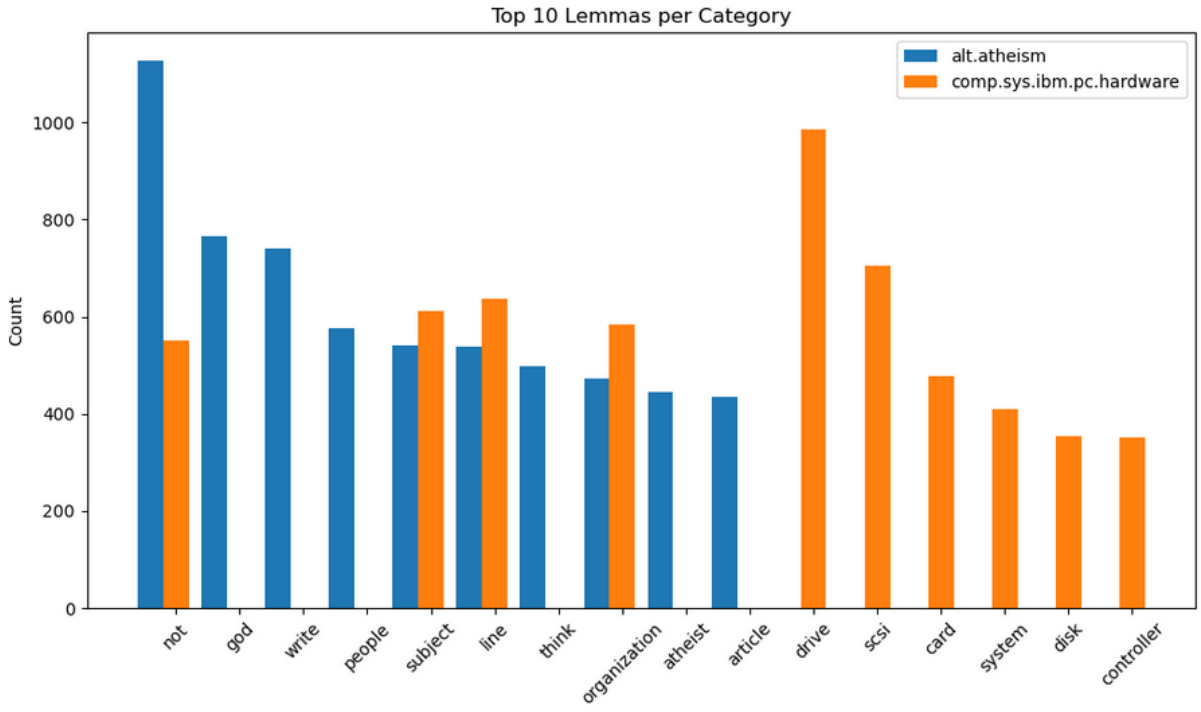


Figure 5: Top 10 Lemmas in both Categories