Stereotypical Beings

Exploring Hidden Classes of Entities using Latent Entity Modelling.

Adam Slack

DRAFT THESIS

Declaration

I declare that this work is my own and not someone elses. May god beat down upon me with hellfire should this declaration be falsly claimed.

Acknowledgements

This thesis would not have been possible without the abundance of support I recieved from family and friends.

Abstract

Latent Entities are approximations of entities found within a corpus allowing for similar entities to be grouped into classes. Through the use of a document segmentation method processing entities and associated terms, LDA can be applied with the intention of extracting topics from each entity segmentation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The Web forms a seemingly ever-expanding resource, with data being generated and information being published at an accelerating rate [2]. As more gain access to the internet, the rate at which new information is made available will only increase. Whether the origins of data be from, social media, news outlets, or e-commerce reviews, much large proportions of the resources on the web exist in a natural language format. From this data in a natural language format there exists underlying information that can be extracted and utilised in the decision making processes of everyday life. Given the amount of processing required to consume data on an international scale, it is necessary to ensure that there exists computational methods that can accommodate for an individual's or a business' needs to understand data. There exist many methods that are effective at extracting surface level information, however there is room for further understanding of hidden structures within datasets, it is by relating distinct pieces of information or subsets of data, that systems can begin to frame or contextualise a solution to a problem within a wider setting. This thesis aims to provide a novel method capable of providing succinct summaries of a set of documents by framing the corpus in terms of Latent Entities.

When considering Information Retrieval (IR), Machine Learning (ML), and Natural Language Processing (NLP) in conjunction with each other, their principal concerns are with the extraction of information from data in the form of natural language texts. Relating textual data yields information valuable to many different entities, including businesses marketing products,

individuals choosing what to read, and even researchers performing analysis on collections of journal papers. For a business, relating entities extracted from texts, can help identify target groups to aim products at. For an individual, relating entities can assist in decisions on what to read, buy, and watch based on similarities between things that they do and don't like. For researchers, the identification of common themes or prominent authors can be achieved through the relating of entities.

A Latent Entity in this investigation is defined as an abstract representation of some entity that can be used to describe one or more concrete entities. Thus the term Latent Entity Modelling is defined as a process of transforming a corpus of text into a collection of Latent Entities describing said corpus. Latent Entities represent a layer of abstraction from Entity Topic Modelling, wherein it is possible to describe collections of Entity Topic Models in terms of extracted Latent Entities.

Entity Topics build upon the concept of topic modelling, which in turn is built upon more simple language modelling techniques. This is particularly when regaring the types of topics that would be derived through Latent Dirichlet Allocation. A Topic in in this thesis is a probability distribution of words, such that the degree of membership for each word in a topic indicates the probability of that word being an indicator of that topic. By performing clustering on extracted Entity Topic Models, this thesis investigates if it is possible to gain benefits in understanding, visualising, and summarising documents.

Oriented around Latent Dirichlet Allocation, Entity Topic Modelling methods have limitations resulting from assumptions made, this thesis addressess some of this through an intermediate pre-processing step allowing larger documents to be utilised more effectively. As an abstract technique, the process of calculating topic models is difficult to visualise, it would be an over-simplification to compare LDA to other classic clustering methods, however the resulting topics would not be to disimilar in structure.

This thesis begins by surveying the field around Natural Language Processing with emphasis on elements related to the task of topic modelling. By highlighting some standard approaches,

their strengths, and weaknesses, it is hoped that the rationale behind the direction of the thesis is clear.

Upon establishing a base of knowledge, new ideas are presented. Major elements include the proposal of adaptations to existing co-occurrence matrix methods, means of deriving Latent Entities through K-Means clustering, as well as metrics for evaluating each of these proposed methods.

Given a set of proposals, the investigation begins by exploring an implementation of a system through which Latent Entity Modelling methods can be explored and evaluated. Such system requires careful consideration of architecture as well effective means of presenting models and data to users.

After discussing the implemented system, a critical evaluation of the modelling methods and general analysis pipeline is presented. Using a mixture of new metrics and traditional ones found in literature.

1.2 Aims and Objectives

The aims of this project are two-fold. The first aim is to utilise and extend natural language processing methods to model and extract latent information describing entities within narratives. The second aim is to develop a system within which corpora can be easily and effectively analysed and explored when methods developed for the first aim are applied to a corpus.

Natural Language Processing is a vast field with many statistical and analytical methods available for modelling purposes, and as such the scope of the first aim is required to be well defined. Objectives for the first aim include:

- To survey the fields of Natural Language Processing and Topic Modelling to identify methods valuable for the task of Latent Entity Modelling.
- To Define a Latent Entity Modelling approach capable of providing models that can succinctly describe entities within narratives.

The second aim involves the development of a software environment, the objectives of which include:

- To develop a front-end user interface for the visual exploration of Latent Entities.

- To implement back-end infrastructure allowing for dynamic visualisations in front-end user interfaces.

- To create a software pipeline that applies preprocessing and extacts Latent Entity data from a provided corpus.

1.2.1 Project Scope

There is legitimate risk of undertaking a project with unreasonable scoping. The aims and objectives for this project apply some upper limit to the scope of the thesis, however there is still room for more discussion on exactly what is and isn't within reason.

Whilst the project aims to consider a range of methods and possible adaptations to them, it is unreasonable to expect any individual to numerically and critically evaluate every proposed method related to Natural Language Processing. It is however within scope to consult recent academic literature surveys that attempt to summarise the current state of the field. By considering field surveys, it is possible then to gain broad insight into what analysis devices are relevant to the aims of this study and more focused probing can be undertaken.

The development of new methods of analysis often require the development of new methods of evaluating proposed methods. There is then a discussion to be held on the validity of those evaluation metrics, potentially leading development into a lengthy iterative process of producing evaluation metrics to evaluate other evaluation metrics. In cases of abstract methods applied to materials deemed subjective, it would not be feasible in this thesis to ensure that the metrics used to evaluate proposed methods are completely free of flaws and shortcomings. Assumptions for the sake of progress would need to be made. When assumptions are made however, evaluating the effect of those assumptions being incorrect would not be out of scope and remains a valuable exercise to undertake.

The focus of the project is on the implementation of Topic Modelling methods, the Latent Entity Modelling approach being part of that aspect. The aim of developing a system capable of automatically processing that corpus is almost naturally going to come to fruition as a side effect of formalised design and implementation of an analysis code base. The interface through

which results and models can be explored however requires additional development occurring entirely aside from a Topic Modelling and NLP toolkit, and thus requires a defined scoping. With regards to visualisations, it is not feasible to produce a complete set of visualisations encompassing all elements of data processed and extracted. Publications are written frequently with a focus on visualising a tiny aspect of natural language data. A more appropriate expectation is to develop a minimum viable set of visualisations capable of demonstrating one or two key points.

With regards to infrastructure, consideration of viable long term hardware solutions to process entire corpora of texts is out of scope. On personal desktops the analysis of a corpus can take days, and producing a system that could accommodate for multiple users analysing multiple corpora is definitely out of scope.

1.3   Motivation

Within social media it has become a seemingly standard practice to profile and analyse individuals on the platform, analytics companies are profiling and classifying individuals for commercial benefit, all whilst the typical user remains completely oblivious. Users are both unaware of the mass social engineering practices that they are subjected to, as well as where they themselves sit within the models developed from their personal information. As the world becomes more data oriented, it is increasingly important that there are ways in which individuals can remain 'data literate'. That is to say, it is crucial to ensure that platforms exist allowing individuals to be aware of the data that surrounds them in everyday life and understand certain aspects of it.

The prevalence of personal data analytics in everyday life is strong motivator to try and produce tools that could help individuals in anyway, many of the methods considered in this paper have applications outside the scope of written narratives, and are used frequently to summarise, relate, and classify a variety of datasets. It is hoped that the proposed Latent Entity Models in this thesis, and the tools for exploring the these models can both allow and encourage individuals to take an interest in the meta-data of their lives.

Chapter 2

Literature Review

2.1 Introduction

Natural Language Processing (NLP) is a vast field of study within Computer Science and Computational Intelligence, the focus of which is the development of intelligent systems capable of handling data in the form of natural language. The task of extracting Latent Entity Models is one that will touch upon many areas of NLP, including Part-of-Speech (POS) tagging, Named Entity Recognition (NER), and Latent Topic Modelling. Given that the focus of this paper is the extraction of a descriptive feature, the applications of similar descriptive features will be considered. Considering the applications of similar features can give potential insight into the value of this thesis.

It is possible to divide NLP into two large schools of thought; One involves the processing of natural language to ensure that machine usable resources are available, the other is concerned with the application of information resulting from processed natural language. Even though the applications of similar modelling methods will be considered, the focus of this chapter will be on methods for producing machine usable resource. Additionally, whilst many NLP tasks require methods for parsing, understanding, or synthesising spoken words, this paper is only considering written texts. As such, the spoken language aspects of NLP will be overlooked,.

Parsing written texts begins with tokenization. This is the division of a single string of characters in to strings representing sentences or words. Often text is tokenized into sentences, and each sentence is then tokenized into words. Once a document is expressed with a basic sentence and word structure it is possible to apply additional processing steps. POS tagging is

the application of tags to sequences of words, each tag and resulting sequence represents the grammatical structure of a sequence of words. NER is the extraction of entities from natural language, entities are tagged depending on their type. NER is performed after tokenization, however it doesn't necessarily rely on text being POS tagged first.

There are varying degrees of subtleties that need to be considered when a process relies on Tokenization, POS Tagging, and NER Tagging methods, failure to consider some of the challenges associated with each method could result in unexpected results in resulting steps. Tokenization is the most straightforward of the three mention processing steps, so only minimal consideration is given to potential tools. POS and NER Tagging are explored in more depth.

A Variety of NLP tasks involve the application of natural language parsing, from Customer relation Chatbots, to Document Summarisation and Relation. Whilst many applications involve the use of parsed text features, many also orient themselves around additional features, For example, tools for relating and summarising distinct documents may utilise topic models and topic modelling techniques.

2.2   Part-Of-Speech Tagging

POS Tagging is a useful task for many NLP problems, it forms a solid platform from which many investigations can be launched. The quality of a POS tagger can make or break a study. The role POS tagging plays in this study is to produce a corpus of words that can be filtered by type. When building a model of words associated with entities and deriving topics that describe entities, there are categories of words that provide much information about entities within a corpus, often words such as 'The', 'To' and 'where' are removed by filtering terms classed as 'stop words', however more flexibility and a greater coverage can be reached by filtering terms tagged as determiners, prepositions, or pronouns.

Whilst removing certain classes of words may see an improvement in the quality of any derived models, it adds additional complexity to the prepossessing aspect of the thesis. Not only do potential methods, and tools need to be considered, but potential tag sets need to be evaluated.

There is a range of POS tagging methods to choose from, many the highest performing taggers employ Maximum Entropy (MaxEnt) models or Hidden Markov Models (HMM). However Rule and transformation based taggers often suffice [3–6]. There even exists hybrid models that use probabilistic or stochastic methods in conjunction with rule sets. UCREL CLAWS tagger is an example hybrid model. [7]

As long as the POS tagging tool utilised performs comparably to those in literature, labouring over the type of POS tagger that is used will not overly affect the results of this study. It might be more useful to consider the subjective merits of any POS tagging libraries that already exist.

2.2.1 POS Tagging Techiques and Tools

The Natural Language Toolkit (NLTK) by default utilises a Maximum Entropy POS Tagger using the Penn-Treebank tagset. The main benefit to this POS tagger is its ease of use and accessibility. The performance of the MaxEnt tagger used in NLTK was reported score an accuracy of 96.64% on all words, and 85.56% on unknown words. [8] However, in similar implementations, certain tags had error rates of 100% , this was likely a result of the absence of certain features necessary to correctly tag specific parts of speech like 'TO' which occured 14,748 times in a study and was never correctly classified [9].

By comparison, the CLAWS tagger achieved a reported 96% accuracy, though accuracy fell to 82% when text was not preprocessed to filter spelling variants and shakespearean english words. The challenges with POS tagging is the variance of sentence structure, as well as ambiguity that can occur within the english language. Fictional narratives like that of shakespeare frequently contribute to the open classes of words (Adjectives, Nouns, etc), meaning that there is an increased likelihood of unknown or spelling variant words. Whilst modern texts are not likely to be as creative with the english language as shakespeare, it would be naive to assume the CLAWS tagger would achieve the upper bounds for its accuracy. Estimating the performance of the CLAWS tagger on the corpora being used in this investigation makes the performance of both the CLAWS tagger and NLTK's MaxEnt tagger roughly comparable.

In much of the literature, POS tagging techniques were evaluated using corpora oriented around

a specific subject or domain, like news, or personal tweets. Using corpora limited to specific domains means that any given POS tagging method might only perform as reported in literature if the same or similar corpus is used. This project will benefit from a POS tagger that performs well on general purpose text, meaning that considering many of the tools used in literature opens a potential point of failure. As POS tagging is one of the first steps performed when processing a corpus, errors could be introduced into the system early on should the POS taggers used in literature proved to be inadequate on the chosen corpus for this project. Additionally, the effect of literature tending to focus on domain limited corpora means that studies on the quality of general purpose cross-domain POS taggers are somewhat unreported.

2.2.2 Tag Sets

When using POS Tagging as a means of filtering certain classes of words, the choice of tag set is arguably the most important decision that needs to be made. By choosing a small tagset, you risk filtering out terms that might actually be important. Alternatively, an excessively expressive will make POS tagging a more difficult task, depending on the method used. For a HMM POS Tagger, training heavily relies on the frequency at which tags and terms occur, using more tags will require a larger training corpus in order for the model to accurately represent written texts.

One of the more restricted tag sets is the Universal tag set, proposed as a potential set of classifications applicable to all languages. [10] As seen in Table 1, the Universal tag set consists of the most fundamental parts of speech, using one classification where many tag sets would use in excess of 5 or 6. The approach taken when developing the tag set was to be pragmatic, and choose tags that cover the most cases, in the most languages. Whether there is a grammar that is truly universal to humans has been a topic of discussion for over half a century [11].

Larger tag sets include that of the Penn-Treebank, and Brown Corpus tag sets, with 36 and 82 tags respectively. Even larger still is are those building upon the Brown Corpus, the London-Lund Corpus of Spoken English uses 197 distinct tags. The larger tag sets are formed with the belief that each distinct tag captures a specific mechanism that can be employed within

Tag	Type
VERB	verbs (all tenses and modes)
NOUN	nouns (common and proper)
PRON	pronouns
ADJ	adjectives
ADV	adverbs
ADP	adpositions (prepositions and postpositions)
CONJ	conjunctions
DET	determiners
NUM	cardinal numbers
PRT	particles or other function words
X	other: foreign words, typos, abbreviations
.	punctuation

Table 1: The 12 Tags Forming the Universal Tag Set.

language. [12]

The Penn-Treebank tag is is by no means a large tag set and by reducing the scope of the tag set, it provides fewer opportunities for inconsistencies to arise. a discussion was held around the validity of the brown corpus given that syntactic context is overlooked, with words tagged independently of their syntactic function. [13] When evaluating which tag set to choose, the corpus being used during the training and application of models is an important factor to consider. Often tag sets are built fo r the purpose of effectively tagging a corpus resulting in effective tagging of that corpus and related corpora, the tag set used with the brown corpus is a prime example of this.

2.3   Named Entity Recognition

The task of extracting a list of entities from text is a similar task to that of POS tagging. It involves parsing natural language and applying entity labels to entity words. A recent survey of the field summarised that Named Entity Recognition (NER) - the task of identifying entities and labelling them as 'Person', 'Organisation', or 'Location' - as being essential to many tasks of computational linguistics [14]. The ambiguity of natural language impacts NER in the same way as POS tagging, meaning that solutions to the problems investigated in NER don't necessarily have a simple solution. A key ambiguity relating to entities is that it is not always clear which entity some text may be referring to, especially in situations where an entity is referred to indirectly, or when entities are named in unusual ways. [15] NER techniques are considered for

Tag	Description	Tag	Description
CC	Coordinating conjunction	RB	Adverb
CD	Cardinal number	RBR	Adverb, comparative
DT	Determiner	RBS	Adverb, superlative
EX	Existential there	RP	Particle
FW	Foreign word	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass		
NNS	Noun, plural	VBN	Verb, past participle
NNP	Proper noun, singular	VBP	Verb, non-3rd person singular present
NNPS	Proper noun, plural		
PDT	Predeterminer	VBZ	Verb, 3rd person singular present
POS	Possessive ending		
PRP	Personal pronoun	WDT	Wh-determiner
PRP$	Possessive pronoun	WP	Wh-pronoun
IN	Preposition or subordinating conjunction	WP$	Possessive wh-pronoun
		WRB	Wh-adverb

Table 2: The 36 Tags Forming the Penn-Treebank Tag Set

this study as a means of extracting a set of entities from which topic models can be derived and explored.

2.3.1 NER Techniques

NER can be carried out using a range of different statistical methods. HHMs can be used to classify entities as either a name (person, organisation, or location), time, or numerical quantity. HMM-based Chunk Taggers have a reported accuracy ranging from 87% to 94% depending on the size of the training set [16].

Similarly to POS Tagging, statistical methods based on Maximum Entropy can be applied with similar levels of accuracy as other methods [17, 18]. Hybrid approach to NER have been investigated, by utilising HMM, MaxEnt and transformation-based learning, error rates were reduced by as much as 15% when used with the english language [19].

CRFs are a form of statistical model that is particularly useful for applying labels to sequence data, when applied to POS tagging or NER, CRF systems can attain error rates as low as 5.55% and 15.96% respectively [20, 21].

It is worth noting that the drawbacks that applied to publications regarding POS tagging, also apply to studies on NER. Notably, NER methods typically only concern themselves with labelling words in text, and provide no means of extracting additional information about a recognised named entity. Drawbacks common though many of the techniques revolve around the resolution of which entities are actually the same. Entities within books can be referred to directly, or indirectly, and even be addressed with different names. Statistical models also lead way to strange inconsistencies particularly in cases where elements of text are labelled as something that to a human is clearly wrong. In text not strictly following the grammatical rules of the language, or creative texts exploring the language, statistical NER techniques alone may prove inadequate, similarly, rule based NER would fall short.

2.3.2 NER Tools

Many free NER tools exist on the web, including Stanford NER, Illinois NER, OpenCalais NER, and Alias-i LingPipe. In a comparison between the relative performances of these tools to classify entity types (Person, Location, Organisation) the Stanford NER performed strongly overall, achieving the second highest precision and the highest recall rates. [22]. In separate comparison of NER tagging tools, more mixed results were received for the stanford NER tool. It performed comparably generalised to NER tools found in the NLTK and apache OpenNLP toolkits. On specialised datasets, purpose specific NER taggers tuned to the task at hand predictably outperformed un-tuned implementations of the Stanford NER.

The Stanford NER tagger has been shown to be a robust and effective tool, frequently outperforming other freely available tools. The Stanford NER Tagger is an NER tool that utilises conditional random fields (CRF) to identify and classify entities, a Java implementation, the tool exists as part of a larger CoreNLP toolkit created at Stanford University [23]. The self-reported performance of the tool ranges from 92.15% to 85.6% and 92.39% to 85.53% for precision and recall respectively. The upper bounds for possible precision and recall relied on additional processing for handling specific features of text. [24]. Additionally, these results are from tests occuring in 2006. Recent advances in CRFs have been utilised in subsequent versions

of the Stanford NER.

2.4 Co-Occurrence Matrix

A co-occurrence matrix is N by N matrix for N words in a corpus, each element of the matrix indicates the number of times two terms have occurred in the same space (often done at a sentence level). A co-occurrence matrix provides a means of quantifying relationships between datapoints and whilst have a range of applications in image processing, they are most useful for this thesis in an NLP context. Many NLP methods utilise and benefit from data represented in a co-occurence format, including probabilistic latent semantic analysis. [25]

A classical co-occurrence matrix is fundamentally reductionist, they operate over a fixed windows and simply count the number of times a term exists with another term in that window [26, 27]. Fuzzy approaches have attempted to capture the degrees of associativity between two terms in the co-occurrence matrix, and whilst achieving promising results, the domain in which investigations were carried out are largely unrelated to NLP. [28, 29]

2.5 Topic Models

Topic models are a way of representing a corpus of text in terms of latent or abstract topics. A topic is defined as the degrees of membership each term in a collection has to a topic. In methods like Latent Dirichlet Allocation, a corpus is expressed as a probability distribution of topics, which in turn are expressed as a probability distribution of words. Describing a specific element of a corpus in terms of topic models requires matching terms in the text to terms in the various topics, after the derivation of a set of models. It is possible to conceptually regard Topic Modelling as a clustering method, where terms are grouped into clusters or topics by considering the statistical likelihoods of specific topics generating the terms present in a document.

Deriving topics from a corpus is commonly done through statistical techniques such as Latent Dirichlet Allocation (LDA), Hierarchical LDA, Latent Semantic Indexing (LSI), or Probabilistic LSI (pLSI). However, many existing methods are in fact built upon the ideas used in LSA,

for example, the valuable LDA method arose out of the shortcomings of the comparatively simple LSI [30]. Many of the challenges around Topic modelling now relate to the human interpretation of developed models. Ensuring that topics are cohesive and interpret able is hard, it often involves restricting data going into the model so that variability within topics can be reduced. Additionally, ensuring models are interpretable is made more difficult by the fact they are not inherently intended for human use, as machine interpretable models, the reasoning behind why topics are structured in such a way is not necessarily clear as a result of their abstract nature.

2.5.1 Latent Semantic Analysis

Sometimes referred to as Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA) is a technique that provides vector representations of documents in a corpus. These vector representations allow for quantitative comparisons of different documents. The implementation of the technique is oriented around Single Value Decomposition (SVD), and attempts to model terms in a document as being averages of the document passages which in which it occurs [31]. Additionally, for NLP tasks It is possible to view LSA as an extension of the term frequency-inverse document frequency (tf-idf) method, this is due to the method producing a reduced set of the same reduction carried out by tf-idf such that term occurrences with the most variation between documents are retained [32].

The obvious shortcoming of LSA is focused purely on documents at a word level, it bears no notion of topics, themes, or higher level concepts through which documents can be related. The assumption made for the results of LSA is that terms occurring in a document are topically related, and that documents containing the same terms must also topically similar.

LSA is wholly insufficient for the analysis of entities within the documents. There is potential for he method to be used as an intermediate step, or perhaps expanded to take into consideration entities with a document and their entity-term associations. A feature of LSA rendering it suitable for this task is the methods ability to perform well with highly dimensional data, and comparably better than standard vector space models like tf-idf. [33] Given its limited nature,

LSA would likely be useful as nothing more Principal Component Analysis style reduction.

2.5.2 Probabilistic Latent Semantic Indexing

Probabilistic LSA (pLSA) is derived as an effort to mathematically formalise the the ideas patented by Deerwester et. al. and the proposed LSA method. The formal statistical nature of the method means that the method can be used in conjunction with other models in a more predictable manner. The probabilistic approach changes the meaning of output produced by pLSA, resulting in models which numerically relate terms to some latent variable.

When used in tasks of prediction, pLSA models outperformed models produced by LSA, at the task of reducing perplexity. Notably, there is a greater difference in performance on more general purpose information retrieval tasks, than on restricted corpora. [25]. This is likely due to the effect that sparsity has on the ability of each method. Whilst still worse than pLSA, LSA actually performed better as the sparsity of training data reduced whereas pLSA's performance decreased. This suggests that the mixture models used in pLSA perhaps overfit as the sparsity of data decreases.

As this study is focused on written narratives, any corpora used will be diverse and intrinsically sparse. As such the overfitting problem potentially reducing the effectiveness of pLSA might not be so much of an issue. However, whilst it does begin to numerically relate variables, there is still no explicit relation between documents expressed within the models meaning that adding additional documents to the corpus would mean the algorithm needs to be re-run, Additionally, there remains no direct means of relating entities or expressing entities in terms of elements within a corpus.

2.5.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) uses mixture models to generate a collection of topics which can be used to describe individual documents within a corpus. Each document is viewed not as a collection of words, but as different proportions of topics, with each topic able generate different terms that would produce the document. By representing individual topics as probability

distributions of words found within the corpus, additional documents can be expressed as distributions of the topics already derived. This approach overcomes some of the drawbacks of the pLSA method, whilst providing a somewhat human-interpretable representation of the corpus. [30]

A Plate Notation representation of the method can be seen in Figure 1. The variables α and β outside of the plates, represent the document-topic, and word-topic Dirichlet priors respectively. The outer plate represents the document selection across M documents, whilst the inner plate represents the word-topic and word selection across a vocabulary of N words. In the outer plate, θ_m represents the topic distribution for document m. ϕ_k represents the word distribution for topic k, whilst $z_{m,n}$ is the topic for word $w_{m,n}$.
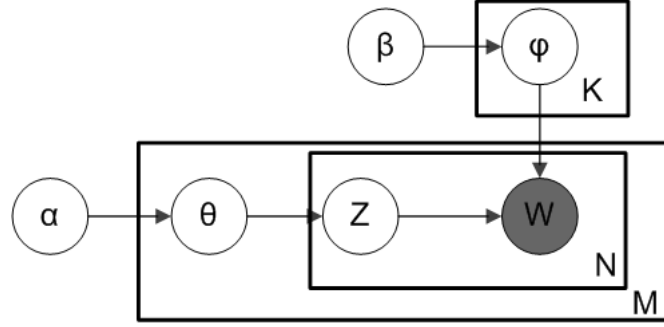
Figure 1: Plate Notation for the LDA Method [1].

In a non-plate notation form, the simplified process for generating an LDA model across M documents is as follows:

$1. Choose\, N \sim Poisson(\xi)$
$2. Choose\, \theta_m \sim Dirichlet(\alpha)$
$3. For\, each\, of\, the\, N_m\, words\, w_{m,n}:$
$\qquad a)\, Choose\, a\, topic\, z_{m,n} \sim Multinomial(\theta_m)$
$\qquad b)\, Choose\, a\, word\, w_{m,n} \sim Multinomial(\phi_{z_{m,n}})$

Where $\underline{W}$ vectors denoting the set of N words across M documents, $\underline{Z}$, is the topic assignment for a word in a document, $\underline{\theta}$, is the topic distributions of each document, and $\underline{\phi}$ is the set of word distributions for each topic.

The total probability of an LDA model is defined as:

$$P(\underline{W}, \underline{Z}, \underline{\theta}, \underline{\phi}|\alpha, \beta) = \prod_{i=1}^{K} P(\phi_i|\beta) \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\phi_{z_{j,t}})$$

In LDA, the lengths of documents are said to be Poisson distributed, and that the set of probabilities generated for each word w_n is a random multinomial dirichlet distribution. There is little reasoning for these assumptions. The use of Poisson distributions in literature seems to be a standard distribution for estimating document lengths. However, given that the lengths of documents are known parameters, the first step in the process is redundant leaving only steps 2 and 3 of the original process.

The versatility of LDA is one of its strongest characteristics. The method has seen use in a range of different settings, from filtering web spam, to the semantic annotation of satellite images. As a bayesian statistical process, it is possible to utilise other methods with relative ease given its modular nature.

In web spam filtering, LDA has been combined with tf-idf and expanded to take into consideration links that exist between documents. For spam classification, the use of LDA as a bayesian network performed worse than baseline machine learning methods, however improvements were seen when the method was expanded and combined with other statistical techniques. [34]

For automatic image annotation, additional machine learning and computer vision methods can be applied to images in order transform images into collections of image features. Topics derived from a corpus of these images can be used to annotate new images introduced. Attempts at using LDA in image annotation have seen some success, whilst not all annotations from topics were correct, they were often at least conceptually related. [35, 36]

LDA doesn't relate words semantically. The meanings of words in a topic are not necessarily related in any way, which is demonstrated in studies using LDA to annotate images. This means that derived topics might not provide any meaningful grouping of words. The intuition is that words in books and documents about a specific subject will somewhat relate to other words in the document. LDA might not produce thematically related topics if the documents used to derive topic distributions are varied in content. For this investigation, the potential for thematically unrelated topics is potentially a major drawback. Entities within narratives might be present throughout a whole series of books. Meaning that they may be associated with a

wide range of topics. It is unclear at this stage what effect this may have on the quality of derived topics. It might be necessary to regard the same entity across two books to actually be distinct.

As a bag of words method, each unigram is treated as being unrelated to other words. Which is not true in the case of written narratives. Much of the meaning of words and relations between words are lost when used in 'bag of word' models. The drawback for this study is that relations between terms and entities are required. If topics were directly derived from the documents which entities were extracted from, then there would be no meaningful way of identifying how these topics apply to entities.

2.5.4 Existing Implementations

LDA is a specialised method of analysis, with only a limited number existing implementations ready and available. The popular python scientific computing library scikit-learn contains an implementation based on open source development coming from Princeton. [37, 38]. Genism is an alternative python library that focuses only on topic modelling methods, it is quite comprehensive in the supporting tools it provides. [39]

The scikit-learn implementation of LDA outputs a set of topics each consisting of a set of word-strength pairs. Figure 2 shows the general structure of output from the method, which is the set of word distributions for the set of topics ϕ. Genism provides a more elaborate interface through which the contents of the topics an other information can be retrieved.

$$\underline{lda} = \{t_1 = \{(w_1 : s_{1,1}), ..., (w_j : s_{1,j})\}, ..., t_i = \{(w_1 : s_{i,1}), ..., (w_j : s_{i,j})\}\}$$

Figure 2: Output format of lda implementation in scikit-learn. where t_i is the topic, w_j is the word, and $s_{i,j}$ is the strength of ownership between the topic and the word.

A key benefit to be had from utilising Genism, is the number of options that are available with the library. As well as a base LDA model there are varying implementations that parallelise the process or utilise proposed heirarchical approaches. Scikit-learn offers a simple method that

provides minimal thrills, however, the simple implementation opens it up for extensions and adaptations in ways that Genism does not.

2.6 Evaluating Topic Models

Topic models are machine representations of word groups that have been deemed to be somehow related. The challenge for evaluating topic models is that it depends on the task which they are being applied to. Directly evaluating the accuracy of topic models is tough, as an unsupervised bayesian method, the probabilities are not interpretable, they are what they are and no ground truth exist. Where produced models can be effectively evaluated is on their efficacy at some secondary task, this could be to predict or estimate topics that might be present in documents not present in the sample used to train models.

2.6.1 Perplexity

Perplexity is a commonly used metric accross topic modelling literature, and is an estimate of how well a probability distribution or probabilistic model predicts a sample. In the literature for pLSA and LDA, perplexity is one of the main metrics used to evaluate the quality of the methods. [25, 30]

Perplexity can be calculated by subsampling a corpus into training and testing sets. By using the test set as unlablled data the perplexity of a model is measured as its ability to estimate the probabilities density of the M documents in the test set.
For LDA, the perplexity is defined as:

$$perplexity(D_{test}) = exp\{-\frac{\sum_{d=1}^{M} log(p(w_d))}{\sum_{d=1}^{M} N_d}\}$$

The probability of a word w_d is seemingly glossed over in many papers, however some methods for estimating the value have been proposed. The harmonic mean is often used as an estimator due to its relative simplicity and computational efficiency, however it has been subject to criticism regarding its suitablity for these Other methods including 'left-to'right' and a chib-style estimator have been shown to be more effective estimators. [40–43]

The use of perplexity as a metric for evaluating topic models does not necessarily mean that the model is correct or accurate, but rather that the model is capable of predicting a sample, not that the prediction is correct. If a topic model assigned equal proabilities accross all words in a vocabulary, then the perplexity would be high, indicating that the topic model provides no insight into the relationships between documents.

2.6.2 Interpretability and Coherence

For human interpretable topic models, it is necessary that the terms within each topic be semantically related, or that it is clear why words in a topic co-exist. Whilst attempts have been made to utilise machine learning methods in the production of coherent topic models, the evaluation of them is somewhat lax. Whilst on the surface, models appear to be more coherent with the inclusion of methods like Markov Random Fields, the coherence is often done subjectively using the opinions of juman judges. [44] Even though the use of human judges may be an imprecise method of evaluation, it does provide insight into the interpretability of derived models. For tools that aim to provide users with insight into large corpora of data, interpretability of the results is a key element defining the success of the tool. With low coherence of topics and thus low levels of interpretability, communicating any extracted information becomes increasingly difficult.

Assessing the interpretability and coherence of topic models has been somewhat formalised through the proposal of a 'word intrusion' and 'topic intrusion' tests. The word intrusion test assesses the coherence of a topic by introducing words that don't belong in a topic and asking humans to identify the misplaced word. A Topic Intrusion test is performed by introducing an incorrect topic and asking humans to identify the misplaced topic. It was found that in cases where topics are incoherent or difficult to interpret, that humans would tend to choose a word seemingly at random. [45]. This method has been extended by asking humans to select two intruder words, where only one of them is actually an intrusion. The idea is that in well defined topics not only will the correct intruding word be selected, but also there will be no means for individuals to distinguish between the strength of belonging for the remaining words.

If participants can't decide on what is an intruding word, then there should be a seemingly randomised selection. [46] Both approaches result in different assessments of any derived topics, there is no indication however as to which metric is a better estimate of the coherence of topics.

The precision metrics defined for word and topic intrusion tests may be of value when evaluating the quality of derived Entity Models, though they rely heavily on human evaluation of topics. Use of judges is time consuming and potentially unreliable. A more formal method of evaluating coherence was proposed by comparing word vectors formed in a semantic space built from wikipedia articles. By considering the distributional similarity of word vectors formed by pairs of words in topics, the semantic cohesion of topics can be estimated using one of a few different metrics. When considering the inter-rater agreement between the calculations and human raters, the spearmann rank correlation values had an average of $\bar{x} = 0.77$ and a standard deviation of, $\sigma = 0.04$. The relative aggreement between each method of evaluation suggests that any of them could serve as valuable means of evaluating the coherence of topics. [47]

Given a topic $T = w_1, ..., w_n$, the coherence of that topic is defined as the mean similarity of each possible word pairing $Sim(w_i, w_j)$ where $w_i, w_j \in T$. The similarity of two words could be one of many measures, including the cosine of the word vectors, or the jaccard coefficient. [48]

$$Coherence_{sim}(T) = \frac{\sum_{\substack{1<i<n-1 \\ i+1<j<n}} Sim(w_i, w_j)}{\binom{n}{2}}$$

2.6.3 Comparing Models

When evaluating topic models, it is often done from the perspective selecting the best topic model. This could be doen through the use of interpretability and coherence tests, or it could be done through a models performance at some secondary task.For exploratory purposes, visual means of comparison could also be used to explore the differences between models and methods.

When regarding topics as a $k - dimensional$ model, visualising the topic space can be challenging. One approach to comparing topic models is to look at the similarities between each topic in a pair of models. In the form of a bipartite graph, it is clear which topics in two models are conceptually similar. The bipartite graph in Figure 3 shows that even though two
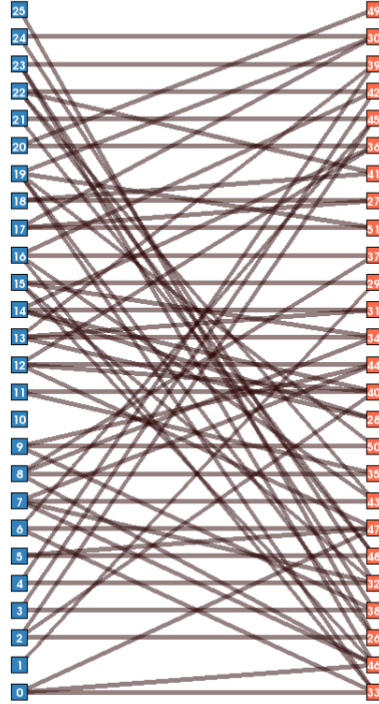
Figure 3: Bipartite topic conceptual similarity graph. (Middle, Right)

methods do have different conceptual topics, there are some topics in both that are somewhat similar. The similarities and differences between the two models are seen further if relationships between documents are plotted.The noticable difference between the two topic modelling methods made visiable in the visualisation is that LDA appears to form relations between the document clusters that LSA kept distinct. [49]

Visualisations might be effective at explaining why a perticular model performs in a certain way at some secondary task, or it might be useful for identifying traits of models that ar valuable for perticular tasks. However, the visualisations don't provide any quantitative insight into how one model may perform when compared to another. The mapping of document relationships seen in Figure 4 also suggest that further classifications can be performed on the documents, whilst LDA provides more interconnections between clusters, there are still clusters which can be used to perhaps classify future documents.
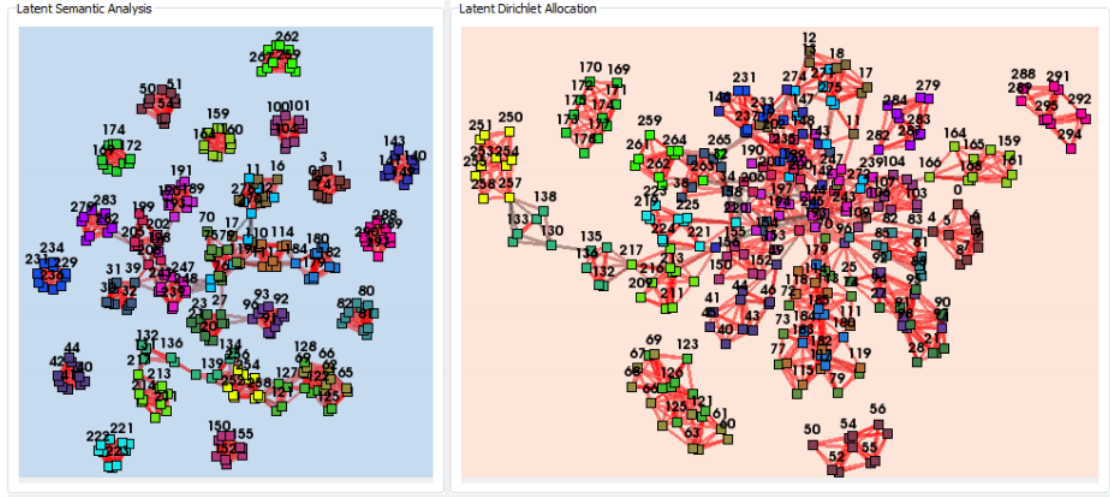
Figure 4: Document relationships formed by LSA and LDA topic modelling methods

2.7 Entity Topic Modeling

Entity Modelling exists as a small field of study aside from topic modelling and could be regarded as an extention of the NER domain. It relates to the extraction of entity description s, as well as the linking of distinct entities. Many of the challenges in this field stem from the high levels of ambiguity present in natural language. Attempts have been made to utilise Topic Models in the task of entitty resolution as well as forming links and relationships between entities within a document, Whilst semantic resources have been investigated as tools for the automatic tagging of extracted entities.

Entity Topic Modelling is often considered for the purpose of Entity linking, which is the task identifying how entities in a corpus relate to each other, It involves the resolution of entity mentions that refer to the same entity in order to perform optimally.

Topic models have been utilised in the task of entity linking through the application and development of specialised Entity-Topic Models.One such investigation involved finding mentions of concrete entities, such as $'iPod'$, $'SteveJobs'$, and $iPhone$ and forming a unifying $'Apple\,Inc'$ topic. [50] The approach added an additional step to the generative LDA algorithm which sampled topic, and entity assignments for each entity. The focus of the paper utilising the topical context within which each entity was mentioned, to try and resolve entities with multiple

mentions through relationships that may form between entities in the model.

The Entity-Topic Models suggested are statistically sound, and perform well in predictive tasks oriented around wikipedia information. The key drawback for this study is the use of short documents to train, test, and evaluate derived models. Models were trained dataset of news articles from the TAC 2009 dataset [51], which being short in length often benefit the assumption made in the study regarding how entities relate to the topics within the document. For longer texts and perticularly narratives, this assumption is not necessarily valid; wide ranges of topics may be present, with entities only being expressed in the context of some of these topics. The limitation of this entity-topic modelling method could come from the base LDA method. As a bag-of-words approach,

Chapter 3

New Ideas

3.1 Introduction

Many different methods of analysis and topic modelling have been considered, with only those specifically discussing the generation of entity-topic models beginning to progress towards meeting the aims that this project has. Of any methods considered, each Entity-Topic Modelling approach had underlying assumptions that rendered them ineffective unless utlised under specific conditions. Additionally, no method directly addressed a need for, or the concept of, a Latent Entity.

The primary contribution of this investigation is a pipeline through which Latent Entity models can be explored. Conceptually, Latent Entity models provide a novel means of summarising, relating, and understanding written narratives. The pipeline propsed utilises a combination of new and existing approaches, with the Latent Entities being extracted using simple clustering methods. As an abstraction of underlying layers of entity topic models, it is necessary to ensure that assumptions made in literature are addressed, else flaws in any developed system ma be further apmplified.

In order to address a fundamental flaw in many aspects of literature on Entity topic models, this investigations proposes the use of a association weighting metric that allows entities to have associations with terms in a document quantified. The Gaussian Entity-Term Matrix outlined in this chapter allows the assumption that entities in a document are equally associated to terms and topics in the same document to be removed.

Evaluating topic modelling methods is a process that requires careful and considerate attention.

It has been shown to be possible to evaluate topic models from a variety of perspectives, including the ability to reduce perplexity, as well as their interpretability. As an abstraction from Entity Topic-Models, typical methods for evaluation may not be applicable, alternatives are proposed in this chapter.

It becomes increasingly difficult to visualise and interpret statistical models as their complexity increases, with the meaning of models becoming obsfucated by the variability of data being analysed. Topic models are no exception to the effect that variability has on levels of obsfucation, and as methods increase in complexity, the need for exploratory tools also increases. In addition proposing a pipeline for extracting latent entities, a system for the visualisation and comparison of topic models is proposed.

3.2   Gaussian Entity-Term Matrix

The Gaussian Entity-term matrix (GETM) is a means of segmenting a given document into entities and associated terms. The matrix for a given document consists of strengths of associactions between a entity-word pairing. It functions under an assumption that the more seperation there is between two words in a document, the weaker the association is between those two words. By modelling documents not as a bag of words, but as sequences it is possible to capture some of the relationships between different words prior to applying additional statistical methods.

Given a set of entities $\underline{E} = \{\underline{e}_0, ..., \underline{e}_i\}$ where $\underline{e}_i = \{e_{i,0}, ..., e_{i,m}\}$, and a set of words $\underline{W} = \{\underline{w}_0, ..., \underline{w}_j\}$, then a matrix D when indexed by i, j can be defined as:

$$D_{i,j} = \sum_{m=0}^{|\underline{e}_i|} \sum_{n=0}^{|\underline{w}_j|} \sqrt{\frac{a}{\pi}} \cdot e^{-a \cdot (e_{i,m} - w_{j,n})}$$

$$
D = \begin{array}{c} \\ w_0 \\ w_1 \\ \vdots \\ w_j \end{array}
\begin{array}{cccc} e_0 & e_1 & \ldots & e_i \end{array}
\left(
\begin{array}{cccc}
D_{0,0} & D_{0,1} & \ldots & D_{0,i} \\
D_{1,0} & D_{1,1} & \ldots & D_{1,i} \\
\vdots & \vdots & \ddots & \vdots \\
D_{j,0} & D_{j,1} & \ldots & D_{j,i}
\end{array}
\right)
$$

This method makes use of the assumption that the more distance there is between two words, or the more words there are seperating two words, the less associated the two words are with each other. There are benefits and downsides to this simplification of the structure of written language. One down side to the assumption made is that, as associations between terms might exactly follow a gaussian distribution, it is possible that the associations between some terms may be exagerated, whilst others are understated. Additionally, the additive approach to handling terms occuring multple times means that there is potentially a need to normalise strengths off associations, this results in additional processing steps that could mitigated by considering alternative methods of handling multiple occurences.

	Harry	Hermione	Ron	neville
loud	0.6	0.5	0.4	
white	0.1	0.2	0.1	
shock	0.1	0.2	0.2	
yell	0.2	0.1	0.3	

"Someone standing outside the Great Hall might well have thought some sort of explosion had taken place, so loud was the noise that erupted from the Gryffindor table. Harry, Ron, and Hermione stood up to yell and cheer as Neville, white with shock, disappeared under a pile of people hugging him." – J. K. Rowling

Table 3: Example GETM for entities within text

As Figure 3 demonstrates, there results in associations between terms that are not necessarily meaningful, the association of Harry to other associations of Harry is not of value, similarly, the association of Harry to words such like 'as' and 'on' is also misleading. As the GETM is only preserving the notion of words having previously held structure and meaning, it is possible to remove perticular parts of speech that don't provide useful insights into the relationships entities have with words. Removing an entity's association to itself is not so straight forward, it requires resolving which mentions refer to which entities, and in cases where entities are either refered to by different names, or two entities share the same name, that is a difficult task. One

solution is to ignore entity-entity associations in the GETM, this is perhaps the best approach given the purpose of the GETM. By removing entity-entity associations from the GETM, latent entity models will describe classess of entities, and are less likely to orient themselves around the relationships between entities.

By modelling associations using a Gaussian prior, it is then possible to adjust the hyperparameter α to increase or decrease the rate at which strengths of association decline as the number of seperating words increase.

It is also possible to remove the gaussian aspect of a GETM, instead utilising different weighting functions. In some texts it might be known that words surrounding entities are associated with strengths following different distributions, thus it is possible to substitute the Gaussian for a more representative prior.

3.2.1 GETM Evaluation

There is no need for hand crafted methods of evaluating the performance of a GETM, or the effect that a GETM has on the ability for meaningful latent entities to be derived. By fixing all values of the GETM to 1.0 (using a fixed weighting function), an entity can be associated with each term in a given document, this would encode the assumptions made in literature directly into the GETM. The fixed GETM can then be used as a baseline to compare different GETMs, whether that be adjusting parameters in the function, or utilising alternative weighting function.

As a result of GETMs being specifically formed for the task of Latent Entity extraction, it is necessary to evaluate from that perspective, the suitability of the method to other tasks would require more generalised evaluation methods. With the availability of a fixed weighting function when calculating the GETM allows for the comparion of quality between calculated Topic moels and Latent Entities with and without the use of a GETM.

3.3 Entity Topic Model

By using a GETM prior to deriving topic models words are already associated with entities, this makes extending LDA topic models a more simple task. As LDA is a bag-of-words method, each entity and its associated terms within the GETM could be regarded as a distinct document. Using GETMs add an additional set of pre-processing steps to the standard LDA algorithm.

The proportional topic distributions in an Entity Topic Model using the GETM can be calulated by multiplying the strength of association between an entity and a given word, with the word-topic probabilities calculated in an LDA model.

$$T_{k,e} = \sum^{w} \phi_{k,w} \cdot D_{e,w}$$

Where $T_{k,e}$ is the Entity Topic for topic k and entity e, $D_{e,w}$ is the strength of assocation between entity e and word w, and $\phi_{k,w}$ is the topic-word probability for topic k and word w.

You may wish to each topic score in an entity topic, bringing values into the $[0,1]$ range. In which case the value for each $T_{k,e}$ where $\underline{T}_e$ is the set of topics for entity e, becomes:

$$T_{k,e} = \frac{\sum^{w} (\phi_{k,w} \cdot D_{e,w}) - max(\underline{T}_e)}{max(\underline{T}_e) - min(\underline{T}_e)}$$

3.3.1 Evaluating ETMs

The evaluation of ETMs can be carried out in a similar means to many topic modelling methods, by considering the reduction of perplexity, or through determining ease of interpretation. The method proposed for calculating ETMs can also give insight into the quality of GETM as an approach.

For this thesis, ETMs are used as a means of obtaining Latent Entities, and should be evaluated from that perspective.

3.4 Latent Entities

A Latent Entity is an abstract statistical model that approximately describes some entities within a corpus. Identifying Latent Entities is possible though clustering entity topic models extracted from a corpus. In the case of k-means clustering, each entity in a GETM matrix is expressed as distributions of entity topics and turned into an Entity-Topic Model. Each Entity-Topic Model can then be clustered in n-dimentional space on topic distributions. The center of each cluster extracted through k-means can then be labelled as a latent entity which abstractly represents entities within that cluster.

As each latent entity remains a probability distribution of entity topics, it is possible to state how similar a given entity is to a latent entity. Each similarity metric of entity to latent entity can then be used to express the entity in terms of similarities to latent entities.

Harry		Hermione		Ron	
Term	Strength	Term	Strength	Term	Strength
Scar	0.1	Magic	0.11	Drooble	0.07
Magic	0.09	Wand	0.09	Magic	0.06
Muggles	0.09	Superiority	0.04	Snigger	0.05
Expelliarmus	0.05	Unkempt	0.04	Sherbert	0.03

Table 4: Example entity-term associations between three entities

Given a set of entities (like Harry, Ron, and Hermione), a Latent Entity for this set could be derived by computing the average entity topic model. Conceptually this latent entity could describe entities that can be classified as 'wizards' or 'students'. The exact interpretion of what the latent entity represents is up for debate, in a similar way to how the topic expressed in a topic model is open to interpretation.

Latent Entities in this form could even be used as a means of assisting in the interpretation of large groups of entity topic models. A summary of ETM clusters can be given allowing for higher level overviews of latent structures within a corpus, or elements of a corpus.

Topics	Harry	Hermione	Ron	Latent Entity
Topic 1	0.6	0.5	0.4	0.5
Topic 2	0.1	0.2	0.1	0.13
Topic 3	0.1	0.2	0.2	0.17
Topic 4	0.2	0.1	0.3	0.2

Table 5: Example entity-Topic distributions for entities in Table 4 and the associated Latent Entity.

3.4.1 Application of K-Means

K-Means clustering is just one approach that can be taken with regards to extracting latent entities from a collection of entity topic models. It is entirely appropriate to utilise other clustering methods, including techniques such as Fuzzy C-Means, Gaussian Mixture Models, or heirarchical style clustering. K-Means as a simple method provides a straightforward approach to finding Latent Entities.

K-Means is a method for minimising variance between clusters and as such can be described as

$$\underset{S}{argmin} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu||^2$$

Given this, the equation for determining a set of latent entities for a given collection of entity topic models involves the substitution of the equation for an entity topic model $T_{k,e}$ into the standard formula for k-means.

3.5 Evaluating Latent Entities

The evaluation of topic models is a challenging task, many techniques commonly used are inherently flawed or difficult to carry out. Many of the flaws present in topic modelling evaluation methods may be made even more pronounced if used to evaluate Entity Topic Models, and Latent. An adaptation of the intruding word test applicable to ETMs proposed here is the intruding Entity Test. Some evaluation metrics are proposed as means of evaluating the quality of latent entities.

3.5.1 Intruding Entity

The intruding entity test involves asking human participants to identify the entity that doesn't belong in a set of entities supposedly represented by a latent entity. The set of entities can be formed by choosing some amount of entities most strongly represente by a Latent Entity, then selcting and adding to the set an entity least likely to be represented by the same latent entity.

Similarly to the intruding word test, the intruding entity test is supposed to performs as a means of evaluating the cohesiveness of the latent entity. Should the set of entities actually be made of those conceptually related, then the odd entity should be identified with a high success rate across all participants.
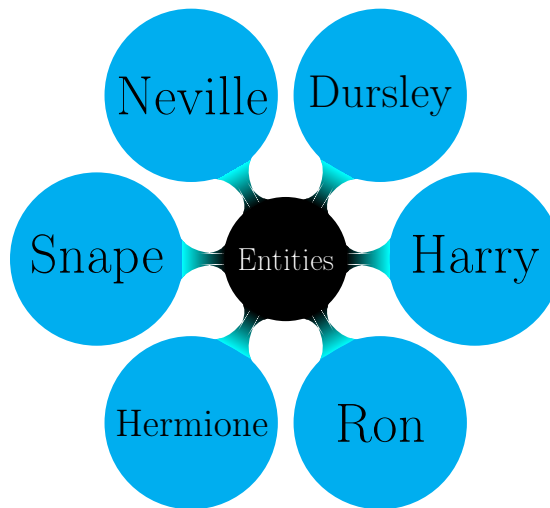
Select the entity that does not belong the the set: _____

Figure 5: Example Set of Entities for the Intruding Entity Test.

In Figure 5 you can see an example of how the test may look, the names of entities being presented in some format, with participants requested to select the entity that does not belong. There are a range of influencing factors that could invalidate the results of any test of this format, most notable of which is the familiarity of the participants to the reference material. In cases where individuals are unfamiliar with the narratives from which entities are drawn from, then there would be increased difficulty when selecting the conceptually unrelated entity.

Admittedly, there is a range of investigations that can be carried out to evaluate the efficacy
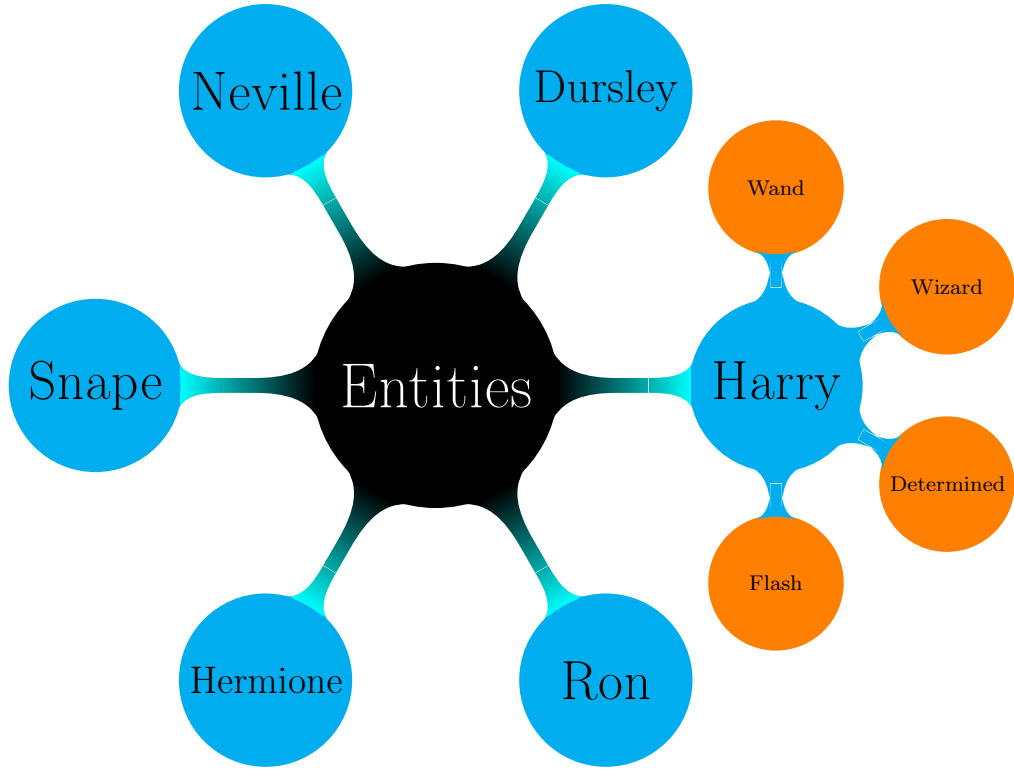
Figure 6: Example Set of Entities with Set of Associated Terms.

of this method of evaluation. It might be necessary to provide participants with additional information about the set in order to allow them to make informed decisions. As the entities within the set are them selves represented by topic models, it is possible to then provide information regarding the words associated with those entities that resulted in the set of entities represented by the calculated Latent Entity.

For participants that are unfamiliar with the matrial from which the entities in the set are from, presenting the terms most strongly associated with the entities might resolve the impact that this could have on their ability to select the correct entitiy. Of course, with the addition of more information the question becomes more complex to answer, there becomes more factors to consider when determining which entity is the unrelated one.

3.5.2 Evaluating Consistency

The consistency of the proposed method for calculating latent entities can be determined by comparing the results of the process for two corpora of different documents onthe same subject.

The assumptions for this method of evaluation is that the contents of books on the same subject are likely to be similar, whilst this may not be true for narratives, factual books on the same subject should hold true.

The consistency check can be used to evaluate different stages of the analysis pipeline. The GETM aspect of the system is a prime candidate for this form of evaluation. Similarly, entities within the GETM should be associated with the same or synonymous terms if assumptions regarding the similarity of two copora are true.

By using copora that are known to be similar in content would remove the perhaps dangerous assumption. However, the corpora are not the same, and thus there still remains the chance that any difference in compared models would be a result of the differences in documents between each corpora.

3.5.2.1   Latent Entity Consistency

For latent entities, it is necessary to consider entities that are present in both corpora that have been clustered differently, and the number of entities absent from one or the other document.

Considering common entities, consistency is calculated using the number of consistencies or true positive (TP), and inconsistencies or false positive (FP) entity pairings within each latent entity clusters. TP is the number of entity pairings in a latent entity cluster present in both sets of latent entities, FP is the opposite of that, the number of entity pairings within clusters not the same across both sets of latent entities.

Where N is the number of Latent entities clusters, and n_i is the number of entities within the latent entity cluster indexed by i, consistency is defined as:

$$Consistency = \frac{TP - FP}{\sum_{i=1}^{N} \binom{n_i}{2}}$$

The defined consistency metric ranges from -1.0 to 1.0. Values approaching 1.0 mean the two sets are increasingly correlated.

Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
a	1	x	a	d	x
b	2	y	1	2	b
c	3	z	c	y	3
d	4	w	2	4	w

Table 6: Left & Right: Entities Within Clusters Behind Latent Entities.

Constencies	(a, c)	(1, 2)	(x, w)	(4, 2)
Inconsistencies	(a, 1)	(a, 2)	(c, 1)	(c, 2)
	(d, 2)	(d, 4)	(y, 2)	(y, 4)
	(x, b)	(x, 3)	(w, b)	(w, 3)

Table 7: Consistencies and Inconstencies Between Clusters Forming Latent Entities

An example of the consistencies and inconsistencies between two sets of latent entities seen in Table 6 can be seen in Table 7. With the total number of pairings, $\sum_{i=1}^{N} \binom{n_i}{2} = 16$, a $TP = 4$ and $FP = 12$ the consistency of these two sets of latent entities is $\frac{4-12}{16} = -0.5$. A negative value implies the latent entiteis are entirely different compositions, and are very inconsistent.

3.6 Analysis Pipeline

The Analysis Pipeline is a series of steps necessary for the task of extracting Latent Entities from a Corpus. Each step can be viewed as a transformation which can be applied to the results of the previous step.

The pipeline is essential to the implementation of a system which can be used to extract Latent Entity information. Without consideration of what the steps would be, the value of any developed system could be severely diminished.

Starting with raw documents, the start of the process is the tokenization of each document in the corpus. The extraction of POS tag information and Entities within the corpus can be

carried out in parallel, it is necessary that both tasks are performed prior to the calculation of the Gaussian Entity-Term Matrix. From the GETM, Entity-Topics and subsequently Latent Entities can be extracted.
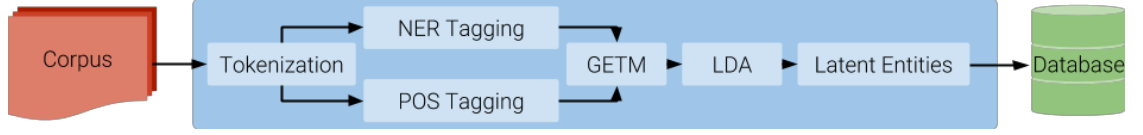
Figure 7: Analysis Pipeline for the Extraction of Latent Entity Information.

3.7 Topic Model Explorer

As machine representations of words most probably related topically, topic models are not always easy to interpret. Presenting only a set of terms within a topic might not provide the best insight into what that topic represents in a corpus. The challenges of interpretation are further compounded as layers of abstractions are added to a topic model. Entity Topic Models and subsequently Latent Entities would be easier to intepret when presented alongside other derived models. As such, a topic model explorer is proposed as a tool allowing the comparison and exploration of topic models, entity topic models, and latent entities.

Humans often try to understand a topic by applying a single word or small phrase as a title of any given topic. Latent Entities are abstractions of multiple Entitiy Topic Models, with each entity topic model being varying distributions of topics and words. In order to accomodate for the human need to label models, the Topic Model Explorer can accomodate for this by demonstrating how different clusters of entity topic models are similar, extracting words most siginificant across members of the cluster.

As well as allowing comparisons of entity topic models, by allowing the explorer to accomodate for the comparison of other levels of topic model extractions, it can allow individuals to evaluate the quality of each model used to describe a corpus.

3.7.1 Interface

The interface for the Topic Model Explorer is a key aspect in allowing individuals to understand the machine models extracted from a corpus. With a poor interface the system is ultimately of no use, users would struggle to use the system and thus struggle to gain an understanding of presented topic models.

The proposed structure of this interface is one that allows users to systematically progress from written text, to information about entities, and then to Latent Entity information. There is essentially four aspects of a corpus that the interface is required to demonstrate for the user to receive a complete picture of the models proposed in this thesis. Book level topic information, corpus level topic information, entity level topic information and corpus level entity topic information.

In a single narrative there can be hundreds of entities, in a large corpus the number of entities is magnitudes higher.

3.7.2 Visualisations

3.8 Project Management

The project began with a highlevel research area, from which an iterative refinement process took place. From the idea of 'Analysing Narratives' general NLP topics were considered, with a focus on understanding the general state of the field. From looking at automatic scene generation, to understanding that even the most fundamental task of NLP (POS and NER Tagging) are still open topics of research, the project was able to focus on one active aspect of the field.

After narrowing the field down to current topics in research, the iterative process underpinned the remaining investigation and development process of the project. Agile development methodologies typically denote how small teams should conduct themselves throughout a software development life cycle, however many of the standard operating procedures of approaches like

Scrum or Extreme Programming can be applied to an individual project.
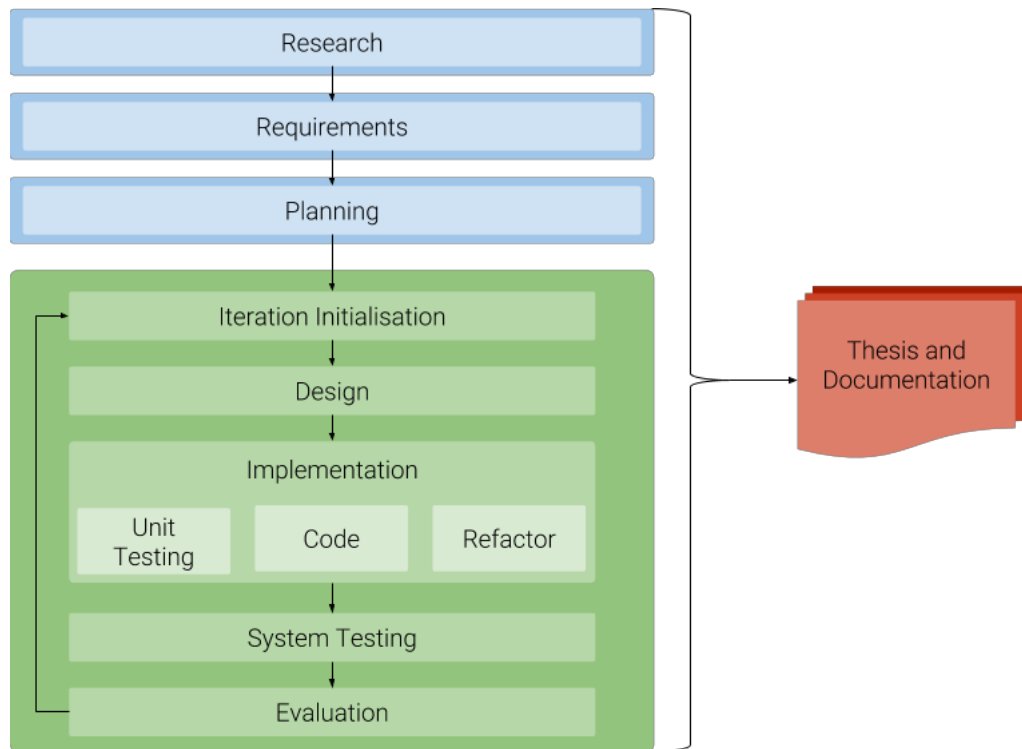
Figure 8: Adapted PXP Process for the development of this Project and Thesis

In a form of personal autonomy, it is necessary to structure development in such a way that focus and direction is not lost, Personal Extreme Programming (PXP) is one such approach. First and foremost, PXP requires discipline and a willingness to learn and adapt, the principals of the methodology are oriented around these two points. [52]

The approach can be adapted to suite a research project of this kind, where requirements and planning are bolstered with an additional 'research' process that accommodates for the investigation into research areas around the topic of the thesis. Additionally, where the methodology contains a continuous process of logging, this investigation instead substitutes logging for documenting. By documenting throughout the development process, the production of the actual thesis is a more attainable task.

3.9 Professional, Social, Ethical, and Legal Issues

With the proposal of methods and tools for the analysis and exploration of data, it is necessary to consider the professional, social, ethical, and legal issues that may be associated with the project as it comes to fruition. Ethical and legal issues are perhaps the most relevant to this project, with potential questions orienting themselves around the morality of encouraging individuals to analyse data, and copyright.

3.9.1 Professional

The development of software aiming to assist an individual in performing a task could have some unintended professional side effects. By reducing the skills required for an individual to carry out analytics task, the barriers to entering certain jobs and industries are also reduced. In critical scenarios like healthcare and security, having skilled and trained individuals is necessary to ensure that recipients of care and those under the protection of others are not harmed in any way, the reduction in workforce quality could impact businesses in this way.

An alternative view of the reduction of required entry levels to positions is that it enables others to perform better generally, either a result of more time being made available, or through the availablity of provided tools.

3.9.2 Social

The social implications of the project relate to those raised in the professional secetion. By providing means by which an individual can effectively analyse and understand a literary landscape, it allows a wider population to engage in writing with an understanding of what others enjoy and find interesting. Opening the doors to new writers of literature could bring numerous benefits to society.

The potential flaw with increasing the rate at which literature is produced is that it becomes more challenging for individuals to find literature that they want and enjoy. It allows the inexperienced and novice writers to structure their documents in such a way that they are

automatically flagged as being relevant to others.

By stroke of the same brush, improving tools for analysis can also assist in distinguishing between subjective qualities of narratives, such that works that aren't very good could perhaps be more easily filtered.

3.9.3 Ethical

Ethically, the principal issue of this project relates to making available tools that could potentially be used to group, relate, and classify entities based on personal data. As the title suggests, Latent entities could be regarded as a form of stereotype, or an underlying classification that can be applied to groups of entities. If applied to personal data, then large groups of individuals could be automatically analysed and processed, with labels being applied to them without the ever realising.

With respect to the analysis of literature, this project could be used to filter and effectively sensor literary works, preventing people from ever coming across certain documents should they not match preferences expressed through a corpus that has been analysed.

3.9.4 Legal

The primary legal concerns of this project relate to copyright and related UK legislation and laws. There is discussions to be had as to what use of an individuals work constitutes as fair usage, even within an academic context it is easy to stray across the line and violate copyright.

For machine analysis, UK Law states that it is considered fair use as long as the project is for an academic, non-profit, and non-commercial reasons. However additional requirements are placed on this in the form of requiring legal access to resources protected under copyright. Fortunately, large collections of texts that are now in the public domain are available through project Gutenberg, an online resource invaluable to academics requiring a large and legal corpus of text.

An additional legal question arises when considering the results of analysis of literature. If

results of analysis form summaries, or can be used to describe a document, then there is

potentially a case to be made about illegal distribution of work protected under copyright.

Chapter 4

Implementation

4.1   Introduction

For an investigation to be carried out into the possibility and efficacy of latent entities, it is necessary to begin with the implmentation of a framework through which a study can be carried out. As such implementation for this project involves the development of 3 key elements:

- Analysis Toolkit
- Web Server Architecture
- Latent Entity Explorer

The Analysis Toolkit involves the derivation and implementation of methods necessary to process and analyse corpora. Web Server Architecture involves the design and implementation of a database, Server, and API capable of allowing access to processed data and results from analysis. The Latent Entity Explorer entails the creation of a set of visualisations and tools allowing for the exploration of Entity-Topic models and Latent Entities.

Once the base framework is in place, implementation turned towards the task of evaluating the proposed process for extracting latent entities, as well as refining implementations.

4.2   Architecture Overview

The system principally follows a standard 3 layer approach to web-server architecture, requiring a database (DB), web server and API, and a client application. This implementation uses PostgreSQL as its DB of choice, as well as following RESTful practices in the API design. The web server is implemented in python, with a DB repository layer accompanying the server and

API implementation. The client application (Latent Entity Explorer) is implemented using TypeScript and the Angular 5 framework.
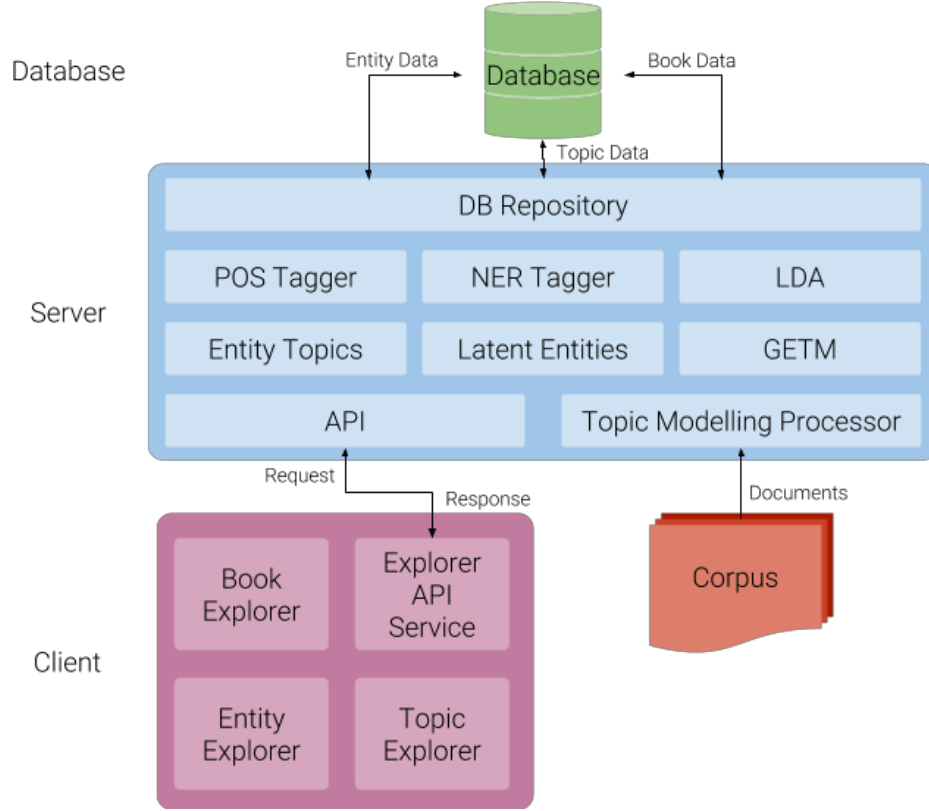
Figure 9: Architecture overview of the Latent Entity Model Analysis and Exploration System.

Alongside the server is a set of Analysis Tools that were developed as distinct methods with utility in their own right.

4.2.1 API

The focus of the API is to allow clients to interact with data resulting from the analysis pipeline. Data to be made available will primarily consist of book meta-data, and extracted information relating to topic models, entity topic models, and Latent Entities.

Given the defined purpose of the API, and the process through which narratives can be analysed, the design of endpoints is a relatively straightforward task. The API will consist of a series of GET request endpoints for book data, as well as a small set of endpoints allowing additional narratives to be added to the system and analysed.

A Subset of the endpoints in the api for requesting data from the system can be seen in Table 8, each of which are GET requests. There are additional endpoints that allow for other aspects of collected and extracted data to be requested, as well as POST endpoints that allow or additional documents to be added to the system.

Endpoint	URI
Book Titles	/api/ books
Book Topics	/api/books/«book title»/topics
Book Entities	/api/books/«book title»/entities
Book Topics terms	/api/topics/«topic id»
Entity Topics	/api/books/«b ook title»/entities/«Entity Name»/topics
Entity Topic Terms	/api/books/«book title»/entities/«Entity Name»/topics/«topic id »
Latent Entities	/api/latent
Latent Entity Topics	/api/latent/«Latent Entity ID»

Table 8: Selection of API Endpoints. Text between guillemots represent variables of the URI.

From Table 8 it is clear how most of the endpoints are oriented around a book title, however, it would not make sense to follow this structure for latent entities. As latent entities are representative of a collection of entities, they are not part of any singular book, and thus are not accessed in the same means as other aspects of the API.

4.2.2   Database

The database is implemented using PostgreSQL, primarily due to familiarity that is held with the DB and library interfaces for various languages (including Python and JavaScript). Additionally, PostgreSQL is an Open Source database implementation, meaning that it is a morally justifiable choice of software.

The database is a key element of the overall system, it is required to provide a persistent means of storing and accessing large amounts of data. There can be vast amounts of data extracted from a corpus, and the amount of time that a process can take to perform the task would mean days are spent waiting for a request to be completed should extracted data not be stored upon

completion of analysis.

4.2.2.1 DB Structure

The Database design was to be oriented around two key concepts: Book Entities, and Topics. Building the DB around these concepts allows for some reflective structure to be maintained between the way that data is stored, and the way that users can access it through the API. The benefit to this is to a developer implementing aspects of software between the database and API, resulting in easier implementation of system logic and analysis tools that access data through the repository layer. Figure 10 shows the relational structure of tables in the database.
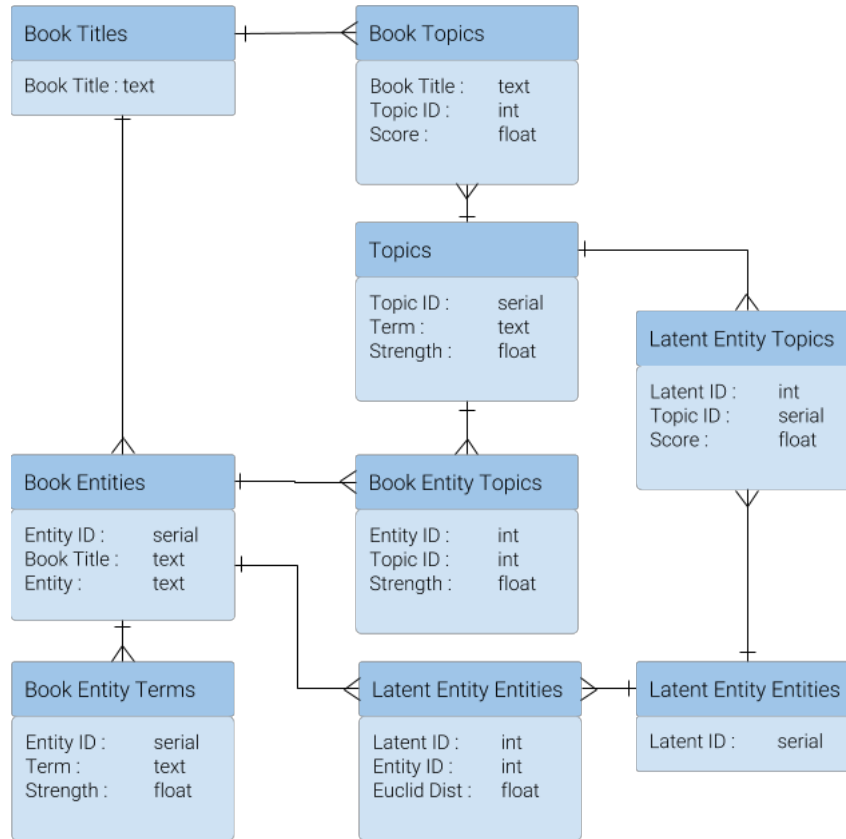
Figure 10: Entity Relationship Diagram Showing the Structure of the System's Database.

A significant table to the system is the 'Book Entity Terms' table, which is a relational database form of a GETM extracted from a corpus. Principally, the table allows for the calculation of Entity Topic Models and Latent Entities to be carried out a later time, whilst also allowing an investigation into how the GETM influences resulting Models.

The tables, 'Latent Entities' and 'Latent Entity Entities' are an important pair for the system, they handle the numerical descriptions of Topics forming each Latent Entity, as well as how each entity in the corpus influenced the structure of each latent Entity.

4.2.3 Server

The Server for this system holds both the API and various distinct processing units forming the system logic. Implemented in Python, the server intergrates well with existing libraries for processing natural language and interacting with databases, it allows for the fast development of a REST API, as well as having a convenient PostgreSQL Library.

The top layer of the server, as shown in Figure 9 is the repository layer. The Repository layer handles all interactions with the database. Below the Repository layer are two layers of processing units with each providing different elements of functionality to the system, from performing POS tagging, to clustering Entity Topic Models.

Below the system logic layers sit two potential interface elements. The first, being the API allows web clients to query and interface with the system, the second is a command line interface element that allows processing of corpora to be initiated and carried out locally within the server.

4.2.3.1 Repository

Using a repository pattern approach for separating the responsibility of interfacing with the database means that a clear structure for inserting, updating, and retrieving data can be defined. In this system, the repository is utilised by both the API and the various processing units and system logic shown in Figure 9.

The methods in the Repository are similar to the endpoints present in the API, meaning that for many of the API endpoints, there is a matching repository method for querying the DB. Table 9 shows a selection of the frequently invoked methods.

Method	Purpose
connect_to_db(host, db_name, usr_name, pswd)	Allows for a connection to the DB to be formed using the provided user details
select_book_titles(db)	Selects book titles from the database using a given DB connection object.
select_book_entities(db, book_title, full_details=false)	Selects entity names in a book specified by book title, selects entity terms if $fullDetails = true$
insert_topic_terms(db, topic_id, term, strength)	a Term and associated strength for a specified topic, as determined using LDA.

Table 9: Select Methods from the DB Repository Layer.

Aside from a single method allowing connections to the DB to be formed, the remaining methods follow a similar pattern as those specified in Table 9. As a result of the purpose of the system, as well as the design of the API, each method orients itself around Entities and Topics.

4.2.3.2 System Logic Layers

The collection of system logic layers consist primarily of methods for carrying out distinct tasks. Many of which satisfy requirements placed on the system by the proposed pipline diagrammed in Figure 7.

Much of the functionality made available to the system is made available at the command line, locally on the server. This is due to the computationally intensive nature of the system, many of the tasks can take days of processing if given a large corpora. A corpus of 7 books could take approximately 3 hours to calculate a GETM for.

Each element of the System Logic Layer explored in further detail when discussing the NLP Toolkit.

4.3 Corpus Pre-Processing

Corpus Pre-Processign involves applying a set of transformations necessary to begin the analysis and creation of descriptive models. It is assumed that the corpus provided to the system will

contain only the text that is to be processed. In many documents there is additional preamble, publishing information, and other elements that can be considered 'noise'. This system will not concern itself with the removal of noise within the corpus.

Tokenization is the first transformation to be applied to the corpus. NLTK provides a suite of tools capable of kickstarting any NLP project, including methods for tokenizing documents. With the option to tokenize a document into sentences, and then words, there are different approaches that can be taken to tokenization. This project uses the 'PunktSentenceTokenizer' methods of NLTK, which is an implementation of an highly effective tokenization method. [53] It is important that the documents are tokenized into sentences, and then each sentence is tokenized into words, as this structure allows for more effective POS tagging.

Once each document in the corpus has sentences and words tokenized, the next stage in the processes is the application of POS tagging methods to determine the gramatical structure of each sentence. A Viterbi Decoded Hidden Markov Model (HMM) POS tagger was implemented in python, utilising the NLP tookit developed for this project. POS Tagging this document means that it is possible to apply filters to the corpus before deriving topic models from the GETM. By removing certain words tagged as certain POS (like determiners, or prepositions) it is possible to derive topics using only descriptive words, which may provide more cohesive and interpretable models.

The defined pre-processing process forms a large section of the overall pipeline, each aspect of the pipeline and preprocessing is implemented in its own distinct elements, each of which have been discussed. In order to tie each separate element into a coherent process, two scripts were developed. One to apply pre-processing at a per-document level, and one at a corpus level.

Figure 11 shows the process taken by the system to apply pre-processing methods to an individual document in the corpus. beginning with tokenization, and ending with the calculation of a GETM for that document. The process of assimilating an entire corpus of documents involves repeating the steps for including one document for a corpus provided to the document. The methods developed for the purpose of implementing this process form part of the NLP Toolkit

which will be discussed in more depth.
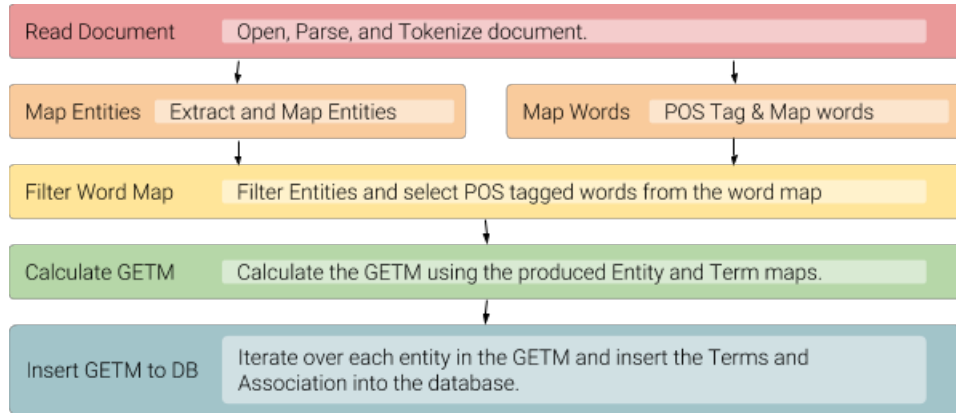
Figure 11: Process taken to pre-process a single document in the corpus.

4.4 NLP Toolkit

As any investigation involving the use of data in any form does, pre-processing was carried out on the collections of documents intending to be analysed. As this investigation aims to also serve as a tool from which multiple corpora can be analysed, the pre-processing steps taken need to be reproducible. The NLP Toolkit developed for this system is essentially a collection of tools that allow each document in the corpus to be pre-processed ready for the extraction of topic models and latent entities. A somewhat functional approach was used, with each method applying some transformation to a single document, or a corpus as a whole. The Analysis Pipeline defined in chapter 3 involves tasks starting with the Tokenization of plain text, and ending with the use of clustering methods to extract Latent Entities.

The toolkit can be viewed as a combination of both simple and complex NLP methods, that allow basic NLP tasks like tokenization and POS tagging to be carried out, as well as more complex GETM and topic modelling techniques.

4.4.1 General NLP

Tokenizing, POS Tagging, NER, an other pre-processing tasks are required as defined in the proposed analysis pipeline. A collection of NLP Utilities allow these tasks to be carried out within the system being developed. Table 10 shows some of the methods developed, as well as

their purpose within the system.

Method	Purpose
sent_tokenize(document)	Returns an array of string tokens representing sentences within the provided document
word_tokenize(sent_tokens)	Returns a mapping of sentence tokens to arrays of word tokens
tag_document(word_tokens)	Returns an array of POS tag sequences for each sentence of word tokens
tag_entities(word_tokes)	Returns an array of tagged entity tuples using NER tagger
cherry_entity_tuples(entities)	Returns a mapping of entity names from set of entity-type tuples

Table 10: NLP Ultity Methods

It is clear that some of the methods defined in Table 10 are relatively straightforward, the method for tokenizing sentences for example, is an important yet relatively self-contained process. In contrast, the methods for applying POS tags and extracting entities are somewhat more involved, with more going on behind the scenes.

As the process behind POS tagging is to be discussed later, The focus here is going to be on how entities are extracted. The NER tooling is not a process purpose built for this project unlike the POS tagger, the commonly used Stanford NER tagger was used to identify entities within text. As explored in Chapter 2, the difference in performance for top NER taggers are somewhat negligible. With the Stanford Tagger performing well in literature, and NLTK supporting its use in Python, it was an obvious choice of tooling.

4.4.2 POS Tagging

A Viterbi Decoded Hidden-Markov Model was implemented to perform the task of POS Tagging. The Viterbi tagger acted as a baseline from which existing taggers could be compared against. Most notably, the NLTK POS Tagger was selected as a potential POS tagging tool, this was a result of how easy the tagger is to use, and how simply it integrates with other libraries.

A dynamic programming algorithm, the Viterbi algorithm provides a comparatively compu-

tationally efficient way of decoding Markov processes, or in this case, find the most probable sequence of POS tags for a given sequence of words. This implementation followed the process described by Jurafsky [54]. The implemented HMM has methods for building a map of transition probabilities between states, calculating the emission probabilities for a given sequence, and calculating the starting probabilities for each possible starting state. Key methods for the tagger are outlined in Table 11.

Method	Purpose
calcStartStateProbs (*taggedCorpus, states*)	Returns a set of probabilities for the likelihood of a sequence starting with a given state, calculated from a tagged training corpus.
calcTransitionProbs (*taggedCorpus, states*)	Returns a set of probabilities for the likelihood of a sequence moving from one state to another, calculated from a tagged training corpus.
calcEmssisionProbs (*taggedCorpus, states, obs*)	Returns a set of probabilities for the likelihood of a given state emitting a given observation.
viterbi (*obs, states, startP, transP, emitP*)	Given a set of Starting, Transition and Emission probabilities, returns the most likely sequence of states for a given sequence of observations.

Table 11: Key methods Forming the HMM POS tagger.

The POS Tagging model formed is essentially composed of the probabilities calculated from a corpus of pre-tagged data. It is possible to regard the methods for calculating the probabilities as being part of a training stage, where the implemented viterbi decoding is only using the model to calculate an answer. By viewing it in this way, there is potential to reduce the processing done by the system by only calculating starting state probabilities, and transition probabilities once. Unless the corpus used to train the POS tagger changes, then the probabilities behind the system won't change. The only time probabilities will need to be re-calculated, will be when determining the emission probabilities for the model. However this is really due to a flaw in the design of the tagger, instead of calculating emission probabilities for a observation sequence, the emission probabilities for all states across the whole training set could be pre-calculated.

4.4.3 Gaussian Entity-Term Matrix

At the core of the process of calculating a GETM for a document, is the weighting equation applied to determine the association between two terms in the corpus. Calculating the matrix involves going through all instances of an entity in a document, and summing the result of the weighting metric using distance between each entity instance and each term in a document.

The process requires a corpus to be separated into two sets, one set of positions for each mention of an entity, and another set for terms. The structure of data expected for the implemented is shown in Figure 12 methods noted in Table 13 require data to be in that structure. There are additional methods allowing data to be put into this standard format.

Entity Occurrence Set	$\{e_0 : \{e_{0,0}, ..., e_{0,j}\}, ..., e_i : \{e_{i,0}, ..., e_{i,j}\}\}$
Term Occurrence Set	$\{w_0 : \{w_{0,0}, ..., w_{0,j}\}, ..., w_i : \{w_{i,0}, ..., w_{i,j}\}\}$

Table 12: Expected Format for Entity and Term Data for calculating GETM.

Method	Parameters	Purpose
Map Entity Occurrence	Entity Array	returns a python dict of entity names and associated position in a document.
Map Word Occurrence	Document	returns a python dict of terms in a document and the associated positions in the document.
Calculate GETM	Entity Map, Term Map, Weighting Function	returns a GETM object calculated from the Entity and Term maps, using the gaussian function as the default weighting function.

Table 13: Methods used in the formatting of data and calculation of GETM.

It is worth noting the format which the collection of entities are in once extracted from a document using the NER Tagger. Stanford's NER tagger produces an array of entities, where each entity consists of three elements, The Name of the entity in plain text, the type of entity that was extracted (person, location, or organisation), and the position of that entity in the document.

4.4.3.1 GETM Objects

A titled-matrix class was implemented allowing matrix calculations to be performed at row or column level using column and row headings. Many of the calculations that would be performed with the GETM occur on a per-row or per-column level on values for terms and entities, and instead of keeping track of the indexes for each entity and term in the matrix this structure allows entity names and terms to be used as keys for each column and row. The ordered map and numpy matrix data structures underpin the structure of this class, with column and row keys in the ordered map linking operations to rows and columns in the matrix.

The index operator for the class allows for cells to be selected by passing either passing the index for the desired entity and term or by passing the string values of both. Additional get and set methods are implemented to accompany the index operator.

4.4.3.2 GETM Visualisation

When implementing the GETM method, a simple visualisation was implemented to allow significant terms associated with an entity to be highlighted. As the GETM was implemented in Python, Matplotlib, a python graph plotting library was used in order to make the task as straightforward as possible.

The implemented graphic is a heat map of the calculated matrix, where a coloured cell indicates the strength of association between entity and term. The darker the cell, the weaker the association. Given the number entity-term combinations occurring in a corpus, and that a GETM can be calculated for a single document at a time, the process for producing the visualisation only considered a single document at a time.

An example of the visualisation can be seen in Figure 12. Even on a single document, there are too many terms and entities for the visualisation to be effective, a general comparison of entities is still possible, with main characters being associated with many more terms. The implementation allows users zoom in and restrict what is shown, increasing effectiveness. Figure 13 shows

a closer inspection on some entities and significant terms.
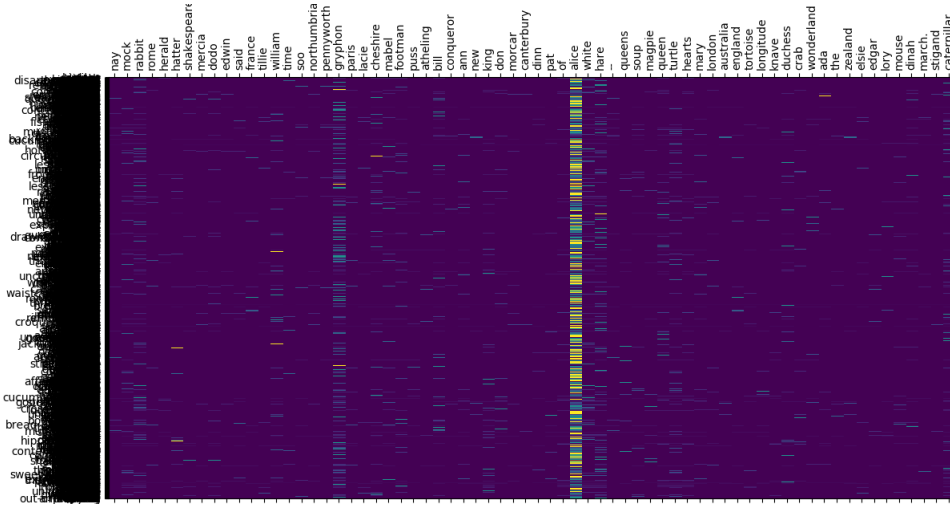
Figure 12: Full Zoomed-out View of the GETM Heat Map for the Alice's Adventures In Wonderland Novel.
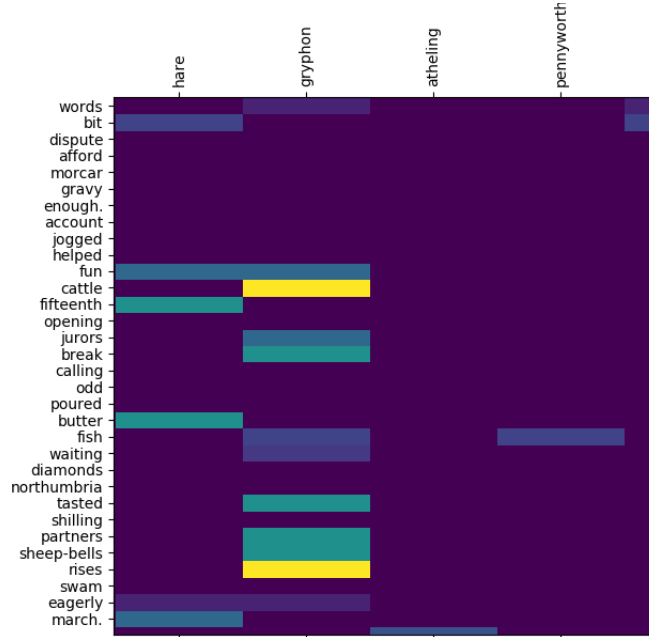
Figure 13: Restricted View of the GETM Visualisation for the Alice's Adventures in Wonderland Novel.

4.5 Entity-Topic Model

Entity Topic models are calculated after the calculation of the GETM. The process is similar to that of calculating topic models for a corpus of documents. Each collection of entity terms is treated as a document which is then fed into the scikit learn implemented LDA algorithm. Like other aspects of the pipeline, this element is also developed in python.

The scikit learn implementation produces word distributions for each topic, in this implementation, the resulting topics can be used to produce entity topic models by looking at the proportional distributions of terms in the GETM for an entitiy, and the score associated with that word across the topics. Topic scores for an entity are normalised in order allow consistent comparisons across entities. Large numbers of Entity Topic Models can be produced in this approach, from which Latent Entities can be calculated.

4.6 Latent Entity Clustering

Clustering was proposed as a potential means of extracting Latent Entities from a collection of Entity Topic Models, specifically the k-means method. The scikit-learn library contains a range of clustering implementations, including a simple k-means algorithm that produces a k-means classifier object.

With Entity Topic Models stored in the system's DB, the process of calculating Latent Entities begins by querying for the model of each entity. After formatting the results of the query using the Numpy library the K-Means clustering process can take place.

The result of the clustering process is a set of center clusters, each of which represent a Latent Entity Model in a format mirroring the structure of the input. Once the Latent Entity Topic Proportions are determined through clustering, a mapping of Entity Topic Models to Latent entities can begin. The euclidean distance in Topic Space between an Entity Topic Model and Latent Entity Model indicate the degree to which an entity is represented by a Latent Entity.

Figure 14 shows a scatter plots for 648 Entity Topic Models. When presented in this format it is somewhat visibile that there are some potential clusters or underlying structures. Significant topics in this form for entities extracted from 3 narratives are Topics 3, 4, and 8.

With only 3 narratives there is the potential for the diversity of topics derived from the GETM to be too low to be of practical use, the lack of diversity manifests itself in the similarity of each Entity Topic Model in Topics 0, 1, 2, 5, 6, 7, and 9. An alternative intepretation is that there is going to be some degree of similarity between entities, and there could concievebly be a

minumim set of similarities that exist between all entities in a corpus. The consequences of this in the implementation is seen when selecting numbers of Latent Entities as well as the number of topics to calculate in other stages.
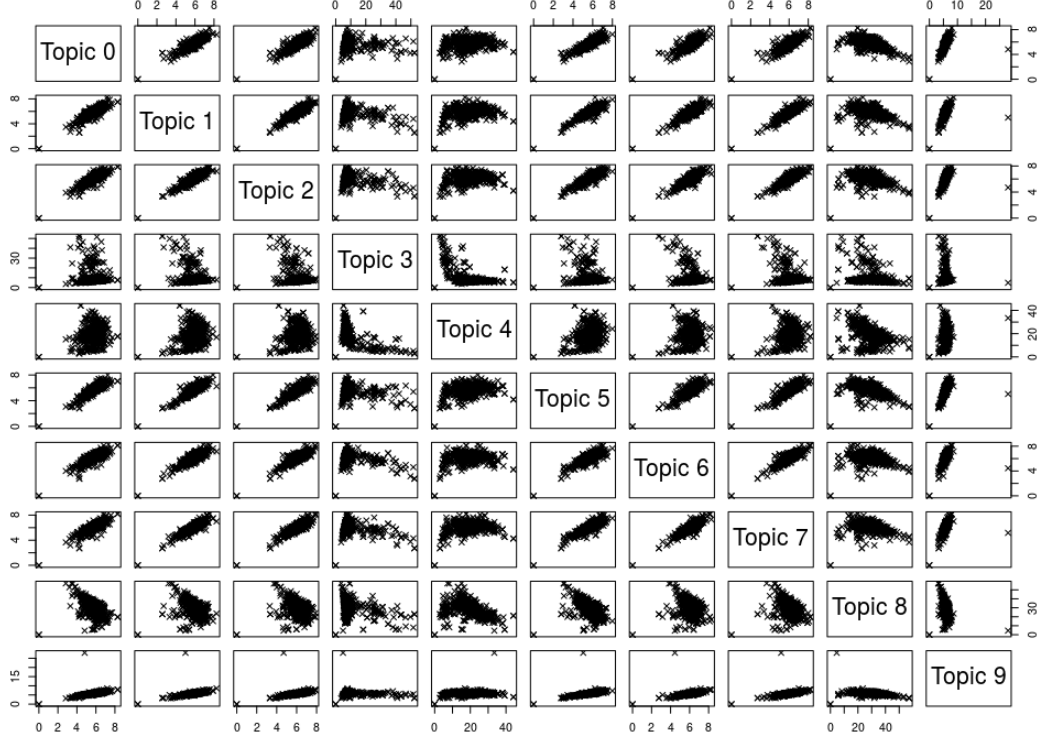
Figure 14: Pairwise Matrix of Scatter Plots for Topic of a Collection of Entity Topic Models.

4.7 Latent Entity Explorer

The interface for the exploring Latent Entities was designed from a personal perspective as a result of its use in supporting the development of other aspects of the system. Implemented using TypeScript and the Angular 5 web framework, the application allows for system to be designed in terms of components, which correspond effectively to the stages seen during the development of the analysis pipeline.

The implemented structure of the application can be seen in Figure 15, the application interacts with the server entirely through a simple Angular 5 service. The Explorer API Service is essentially a singleton class which each other component has access to, it makes available methods necessary for issuing requests and receiving responses and does so using an Observer/Subscriber

approach.

There are 4 main components for the application, three of which are controllers involved in allowing users to explore books, topics, and entities. The fourth Controller is a higher level component that co-ordinates across all elements of the application. In addition to each controller is an equivalent view that allows the user to interact with that aspect of the system. The Explorer View is composed of the other views of the application, when users interact with UI elements, it is through the Explorer View.
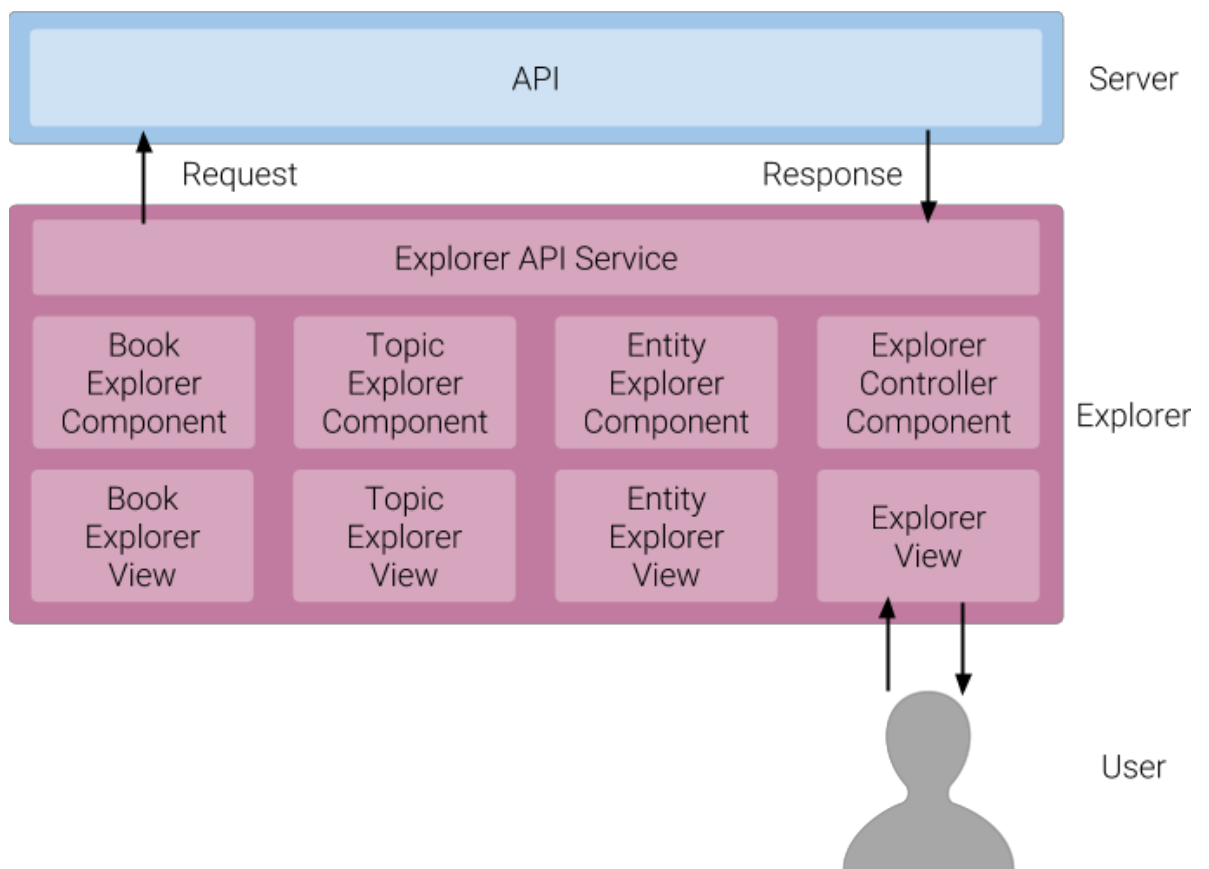
Figure 15: Structure of the Web Application.

4.7.1 UI and Navigation Design

In order to provide a succinct and mobile friendly interface, each element of the system is thought of in terms of components and panels. Each panel provides access to a distinct feature, be it a visualisation of topic distributions in the corpus, or the comparison of entity topic models. Each panel is associated with a component that acts as a controller for that UI

element. Though this thesis is not a study of effective UI design, having a usable interface assists in communicating the results of analyses.

Figure 16 shows initial layout and structure intended to guide the UI of the application. Each panel is of a fixed size, and shows information about a specific aspect of the corpus. Listings of book titles, entities, and topic terms will all be held within separate components. The interface read from left to right, top to bottom, and as progressing though the application, different elements are required to be selected so that more information can be displayed.

Latent Entities are models describing the entire corpus, so wouldn't require the user to specify any information. In order to view Topic Models and information about individual entities in a book, the user would be required to first select a book, and then select an entitiy to view information about that entity.
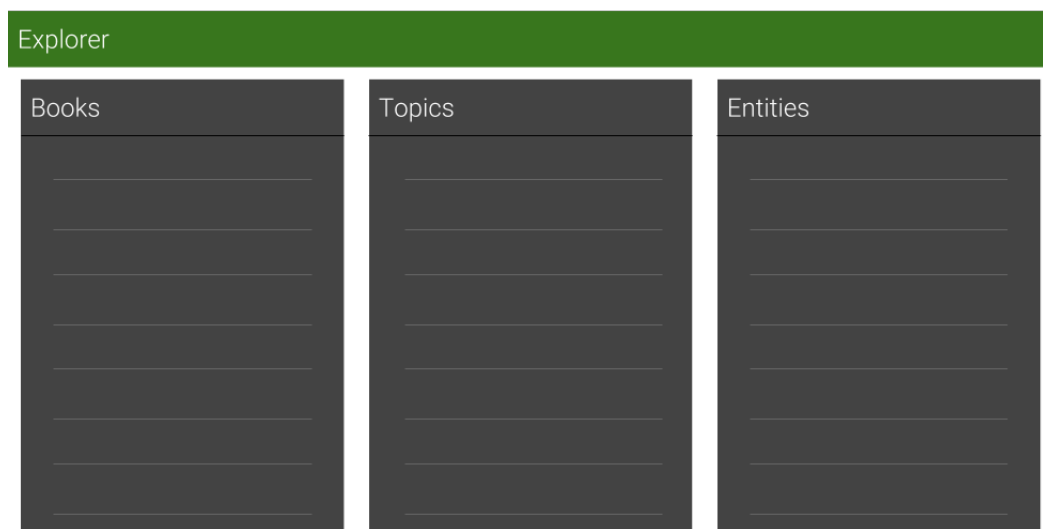
Figure 16: UI mockup layout.

4.7.2 Book Explorer

The Book explorer provides an interface for searching and selecting books analysed in the system, so that data and extracted information resulting from the analysis pipeline can be viewed. Split into two elements, the controller for this part of the application allows users to search for books that have been analysed, as well as initiate requests for book information through the API Service when users select a book. The view elements of the Book Explorer

the follows the panelled style outlined when discussing the UI and Navigation Design.

Delving into details of the corpus begins with the Book Explorer, for users to view entities and thus information about entities, they must first select a book from the listing of analysed books. Upon selection, the component emit's the book selection, which is recieved handled by the acompanying Topic and Entity Explorer components. In addition to emitting events to other component, the Book Explorer invokes method of the API Service for making requests to the server.

Visualising information at a book level is not of particular interest to this investigation, as such the only visualisations associated with each book is actually of the topic distributions and is handled in the Topic Model Explorer component. It is however through the book explorer component that user can signal to other components which entity and topic information it is they wish to view.
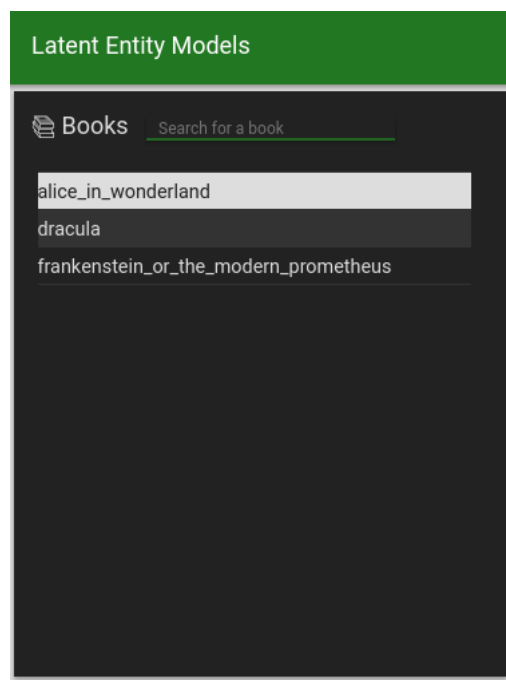
Figure 17: Book Selection Component for Navigating the Explorer.

Figure 17 shows the panel for selecting books, once the user selects a book, a collection of entities and the numerical topic distribution for the selected book is requested from the server. The search operates by making a request through the API Service for book titles containing

the entered string.

4.7.3   Topic Explorer

The Topic explorer is a key component of the application as it provides book level topic information extracted from a corpus, some information about the topics extracted using LDA, as well as the topic distributions for a document selected in the book explorer. Additionally, there is an exploratory visualisation for comparing the topic distributions of each document in the corpus.

The Book-Topic Distributions are presented in the form of a stacked bar chart, where each block on the stack represents the proportion of the document that a topic represents the document. Each colour in the stack represents a different topic, with the same colour representing the same topic across each stack for each document, the panel in the application for this visualisation is shown in Figure 18.
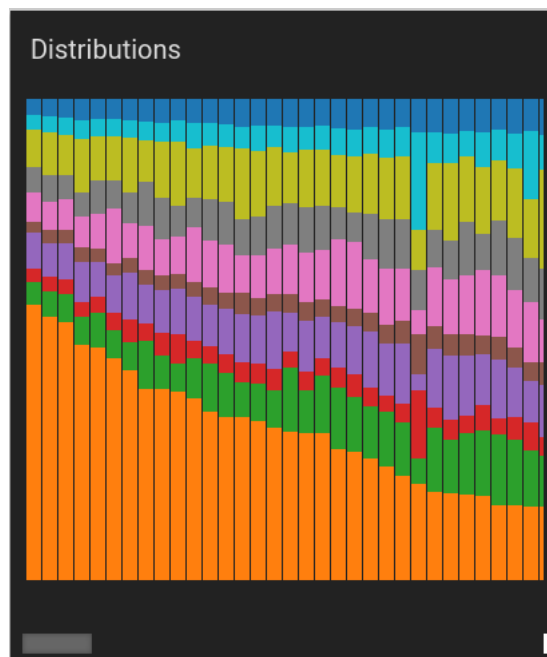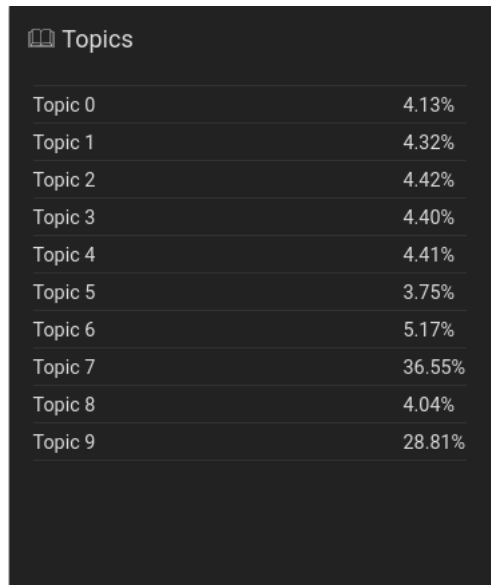
Figure 18: Book-Topic Distribution Panel for Comparing Topical Makeup of the Corpus.

The construction of the chart is done dynamically, making a request for the topic distributions of analysed books in the system. For larger copora, the application would issuing requests and producing the figure for 100,000's of documents, however for this prototype and with

complete control over the size of the corpus that is not an issue that would be encountered. As the visualisation is data-driven using the d3.js library, it is possible to interact and inspect elements of the figure. Users can sort the graph from highest to lowest on topic distribution. Meaning that users can see which book is most represented by each topic. Presenting topics in this way allows for quick visual comparison of documents and topics, as well as providing insight into how a documents topic structure may have an effect on the Entity Topic Models topical make up.

When a document is selected from the Book Explorer, this component uses the emitted book title and the API Service to make a request for the numerical topic distribution of the chosen book. The percentage proportions of topics in the document are shown in a panel in a listings format, and allow individuals to compare Entity Topic proportions to the Document the entity resides in. Figure 19 shows what the user would be presented with.

Figure 19: Book-Topic distributions for a Selected Book.

4.7.4 Entity Explorer

The Entity Explorer Component handles each panel associated with Entity information. Including the selection of entities for further inspection and comparison with other entities, as well as any visualisations for an entity.

When an entity is selected in the Book Explorer component, the Entity Explorer component is signalled that the user has made a request for more information, the component then makes a request through the API Service to the server, requesting entity information for the selected book. Figure 20 shows a view of all entities extracted from the selected book, users can click on an entity to view the Entity Topic Model of that entity, or press the green addition box to add it to to a entity comparison panel. Similarly to the book listing panel in the Book Explorer component, users can search for entities using text to filter entities not containing the query string.

When selecting an entity to be added to the comparison view, the entity is not removed from the listing, and can still be selected for a more detailed inspection if desired. The entity cannot however, be added to the entity comparison panel if it is already present. The Entity Explorer component maintains an array of Entity JSON Objects to allow the tracking of inspected entities.
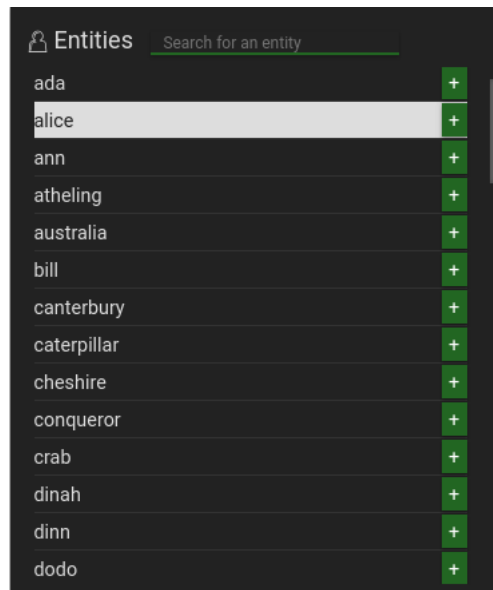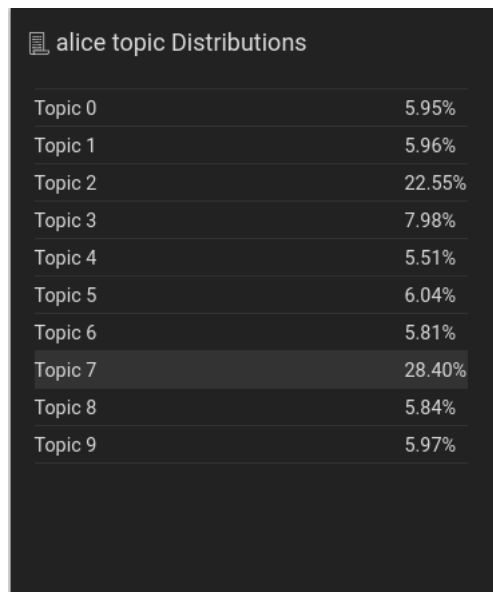
Figure 20: Entity Listings for the Selected book (Alice's Adventures In Wonderland)

When an entity is selected for further inspection, users are presented with a breakdown of the Entity Topic model and GETM results for that entity. Figure 21 shows the topic distributions for the selected entity 'Alice' from Alice's Adventures Into Wonderland, the significant topics for this distribution are Topics two and seven. A visualisation of this distribution, would not

be so informative given the position of the panel in relation to the other panels. Users can compare the percentage proportion of a topic in this panel to the percentage proportion of the topic in other components.

In order to provide some user insight into terms associated with an entity that contribute to an entity's topic model, a panel was formed listing the terms and their strength. Figure 22 shows the most prominent terms for the entity 'Wonderland' in Alice's Adventers in Wonderland. As entity terms were filtered when calculating the GETM, the term 'wonderland' associated with the Entity 'Wonderland' are not actually referring to the same thing.

Figure 21: Entity Topic Distributions for Entity Topic Model.

The panel for comparing entity topics is shown in figure 23, which is the primary interface through which users of the system can begin to compare the topic models that represent an entity. Where entities were added using the '+' in a green box on the entity listing panel, entities can be removed from the comparison panel by pressing the '-' in a red box.

Figure 22: Top Terms Associated with Entity resulting from GETM.

Figure 23: Entity Comparison Panel for inspecting Entity Topic Distributions

Chapter 5

Results and Evaluation

5.1 Introduction

This chapter serves as an opportunity to numerically explore and evaluate the methods and models that have been implemented and discussed in previous chapters. Each aspect of the system is to be evaluated from the perspective of their effectiveness at extracting meaningful information from a corpus. Four key aspects to the system will be quantitatively and qualitatively assessed, including the implemented POS tagger, Gaussian Entity-Term Matrix, GETM Entity Topic Models, and Latent Entity Models.

In order to evaluate many of the defined methods, a corpus must be constructed. As such, this evaluation also explores the structure of the corpora intending to be used to evaluate the whole pipeline. The selected corpus will be also be compared to a larger corpus of documents, this will provide an estimate as to whether the chosen corpus is an appropriate for the task of evaluating the system.

By exploring the results of the implemented methods on the chosen corpus, it is possible to gain insight into the value and worth that the models may have to individuals looking to understand corpora of data that surrounds them.

5.2 POS Tagger

Being a key part of the pre-processing aspect of this investigation, it is necessary to consider the impact that the choice of POS tagger may have on the system. If the implemented POS tagger was not as performant as seen in literature, then it could be used to explain drawbacks within the system.

Unlike Topic Modelling, evaluating the performance of a POS tagger is a more straightforward task. The implemented tagger was evaluated on the Brown corpus, using an 80:20 division of the corpus into training and test samples. Accross both subsets, the corpus contained a total of $57,340$ sentences and $1,161,192$ words, forming a sizable dataset from which a model can be trained.

5.2.1 Results

An overall accuracy of 0.73 was achieved for the method, which is a significantly higher score than randomly applying tags would score, however it is still noticibly lower than scores seen in literature. As the purpose of POS tagging in this thesis is for the filtering of specific types of tags, it is necessary to consider the effectiveness of the tagger on different POS tags.

The tagger classified almost all Verbs in the test set, however when it missclassified parts of speech, they were often missclassified as verbs, which natrually increases the number of successful tags. Notably, the tagger never tagged something that wasn't punctuation, as punctuation, however it failed to correctly label all punctuation in the test set, missing approximately 30%. Table 14 shows a breakdown of the performance of the POS tagger for each posible POS class, it shows the number of times a tag was correctly, and incorrectly applied to a word, as well as the actual number of words with that class in the test set. Finally, Table 14 also shows the percentage of correctly classified words for each pos tag, as well as the percentage of words that were incorrectly labled as that tag.

It is concerning to see that 85% of the terms tagged as 'NUM' were actually something else

as it is one of the classes that would be filtered from the GETM. For almost all casses of misclassification, terms were incorrectly labelled as being a Verb.

One possible explanation of the missclassification is a result of the basic nature of the tagger, when the system has not encountered a word before, it is unsure of what to tag it as. It turns out that in the implementation, the tagger's default classification is a Verb. An interesting feature of the brown corpus is that roughly half of the vocabulary are hapax legomena, leading to a high liklihood that the test set would contain terms unseen to the model.

POS Tag	Correct (n)	Wrong (n)	Actual (n)	Correct Class (%)	Misclassified (%)
NUM	364	2087	487	0.75	0.85
ADV	3747	442	5357	0.70	0.11
PRON	6097	185	7353	0.83	0.03
ADP	6337	420	9630	0.66	0.06
DET	7191	53	10113	0.71	0.01
X	5	6	124	0.04	0.55
NOUN	10606	365	17692	0.60	0.03
ADJ	3378	301	5492	0.62	0.08
PRT	2438	282	3446	0.71	0.102
.	10870	0	15505	0.70	0.00
CONJ	2258	16	3326	0.68	0.01
VERB	16585	21409	16917	0.98	0.56
Total	69876	25566	95442	0.73	0.27
$\bar{x}$	5823.00	2130.50	7953.50	0.66	0.20
σ	4903.48	6097.28	6117.22	0.22	0.29

Table 14: Table of POS Tagging Results

5.3 Evaluation Corpora

Before starting any evaluation involving natrual language text, it is important to consider corpora that may be suitable for use in analysis. This thesis considers a pairing of copora where one consists of a series of novels, and the other consists of summaries of the opposing novels. The same corpora will be used in the evaluation of both the GETM, and Latent Entity models. By comparing summaries of novels to the full books, there is potential for insight to be gained regarding the consistency and ability of the models when used in a document summary context. In order to comply with UK Copyright law, books could only be considered

as candidates if they were legally available to the author of this thesis.

Whilst only forming a relatively small corpus, the Harry Potter series by J. K. Rowling were selected as the narratives through which the evaluation of the thesis could be carried out. This is for numerous reasons, firstly, it is known that there are characters present throughout the series, as well as some that arent. By having entites that aren't ubiquitous it is possible to gain insight into why there may be differences in any consistency tests. Another important factor in the use of these documents, is that they are familiar to the author, and thus a qualitative insight can be provided when evaluating the interpretability and cohesion of the models.

5.3.1   Harry Potter Corpus Statistics

The Harry Potter corpus contains 1,414,420 words ($\bar{x} = 202060$), with a total of 3275 entities extracted from all documements. Topic distributions of the Harry Potter novels are shown in 15, knowing these distributions will show if there is potential for differences in entity topic models about entities common through the series. Figure 24 visually shows some of the differences between the documents in the corpus.

Notably, there is only significant variance in 3 out of the ten topics accross the novels. which manifests itself through the changing distributions shown accross the stacked bar chart. whilst the other topics do see noticable changes in proportions, none are quite as significant as topics 1, 3, and 8. Which on a corpus of related documents, should not be overly suprising. For comparison, figure 25 shows topic distributions for a corpus of over 300 narratives, including the harry potter novels which are highlighted by the black rectangle. The variation between the novels is even less when viewed in the context of a wider corpus.

One of the factors influencing the distinct similarity between narratives in the larger corpus is the inclusion of entities as terms when calculating the LDA topic models, for the reduced corpus of harry potter novels, the entities have been removed. When entities were left in as terms for calculating topics, the method distinguished between novels based on the characters, as opposed to any differences in writing style. With entities left within the calculations, Harry Potter and The Prisoner of Azkaban as well as Harry Potter and the Order of the Phoenix

77

	Topic (%)									
Book	0	1	2	3	4	5	6	7	8	9
Philosopher's Stone	5.16	19.62	5.45	22.29	4.94	5.09	7.04	5.31	20.15	4.96
Chamber of Secrets	4.77	19.73	4.83	15.51	4.54	4.91	10.80	5.02	25.06	4.82
Prisoner of Azkaban	4.33	33.30	4.27	14.21	4.01	4.40	5.50	4.22	21.61	4.16
Goblet of Fire	4.68	19.82	4.67	16.09	4.31	4.69	5.94	4.71	30.60	4.48
Order of the Phoenix	4.66	25.97	4.54	16.07	4.40	4.68	6.22	4.72	24.08	4.67
Half-Blood Prince	4.77	20.05	4.58	17.20	4.35	4.69	6.04	4.78	28.92	4.62
Deathly Hallows	4.88	18.61	4.58	23.68	4.47	4.82	6.27	4.80	23.27	4.62
$\bar{x}$	4.75	22.44	4.70	17.86	4.43	4.75	6.83	4.79	24.81	4.62
σ	0.25	5.37	0.37	3.63	0.28	0.22	1.81	0.33	3.77	0.25

Table 15: Topic Distributions for the Harry Potter Novels by J. K. Rowling
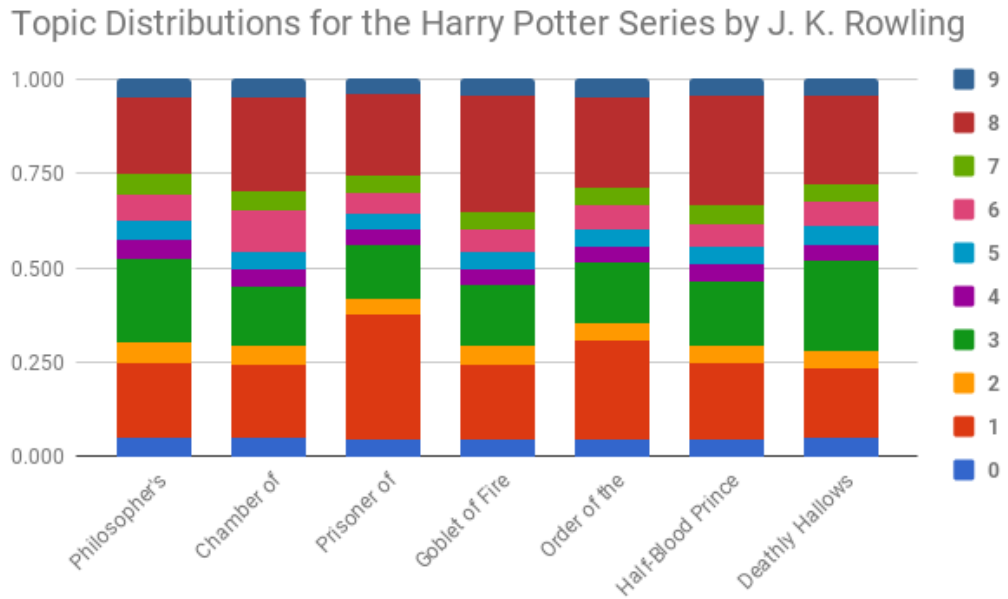
Figure 24: Propotional Topic Distributions for the Harry Potter Novels by J. K. Rowling

were indicated as being very similar documents. When inspecting the topic word distributions, the topics most strongly representing these documents were made up of terms like 'sirius' and 'black', which are referring to an entity prominent in both documents.

Without considering entities within the documents, the distributions are much more similar, however there are still differences between them. Some degree of similarity within documents is expected, given that they are from the same series, and oriented around the same sort of stories.

Key terms in the Topic-word distributions can be seen in table 16, considering the terms that

form the topics might provide some insight into why some documents in the corpus deemed so topically similar. However with topics-word of 1000 terms, viewing a small portion might not provide accurate insights.

0	1	2	3	4	5	6	7	8	9
diary	elf	tent	dementors	elf	elf	tent	elf	elf	dementors
heir	apparently	elf	elf	tournament	map	dementors	lesson	task	firebolt
elf	dementors	locket	remained	map	remained	elf	surface	tournament	lesson
prefect	homework	goblins	hastily	task	tent	ought	task	prime	dementor
attacks	ought	horcruxes	apparently	dementors	task	lesson	ought	lesson	homework
clicking	position	ought	triwizard	apparently	locket	prime	gazing	apparently	map
toilet	lesson	dementors	highly	ought	ought	task	elves	hastily	gazing
petrified	tone	vault	tent	triwizard	dementors	apparently	map	map	elf
camera	thoughts	apparently	thoughts	troll	lesson	locket	dementors	triwizard	ought
wife	ropes	remained	ought	lesson	conversation	thoughts	prime	remained	rat

Table 16: Top Ten Words in the Topic-Word Distributions of Calculated Topics for the Harry Potter Corpus

Many of the topics are actually very similar with regards to the terms which they contain. The term 'elf' is ranked within the top ten for all topics. Whilst the term 'Ought' occurs as a top ten in eight topics. This not only highlights the significance of many of the terms, but also suggests a lack of diversity in the vocabulary for each document. Some terms however are not present through each novel, but still remain present across multiple topics, including 'triwizard' which is prominent only in 'Goblet of Fire'.

With regards to the coherence of the topics, there is little semantic relations between the terms of topics, however within the context of the novels, some relationships can be identified for example, the terms 'goblins', 'vault', 'locket', and 'horcruxes' are all related within the novels. Within the Harry Potter universe, the banking vaults are managed and guarded by goblins, whilst a significant item often simply referred to as the 'locket' was a horcrux. Additionally, a horcrux was also stored within a vault guarded by goblins.

5.3.1.1 Large Corpus Topic Distributions
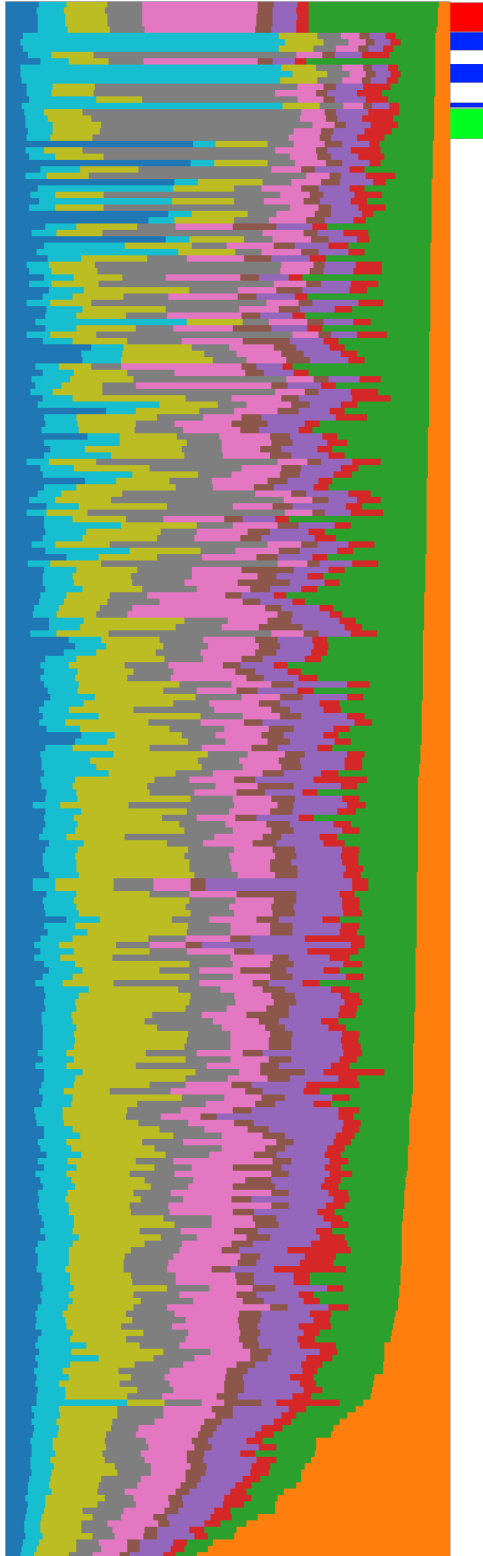
Figure 25: Book Topic Distributions for a Large Corpus of Fiction and Non-Fiction Books

When you consider the topical structures of large corpus of documents, you can see how there are groups of documents have distinctly similar topic distributions, perticularly when ordered by the contribution that a topic makes to a documents distributions. Three document groupings are highlighted in figure 25, where each colour represents a topic and each line represents a document. Larger blocks of colour indicated a larger contribution that a topic has to a given document distributions.

The set of rows indicated by the red block are books written by the author G. R. R. Martin, and include the novels behind the popular TV show 'A Game of Thrones', The the rows indicated by blue are the Harry Potter Novels that are used to evaluate other aspects of the project, the rows indicated by the green tag are narratives written by Agatha Christie.

It turns out that for this large and diverse corpus, the topics that have been extracted are oriented around narratives that exist as part of a larger series, or books that are similar in subject matter. Whilst Topics 5 (pink), 6 (grey), and 8 (cyan), are essentially topics describing books written by G. R. R. Martin, Agatha Christie, and J. K. Rowling respec-

tively, other topics are strongly indicative of writings by other authors.

Topic 3 (red) is a relatively small portion2 of all narratives, however the books that this topic represents the most are all novels by Dan Brown. In contrast, topic 0 (orange), which represents a large proportion of many narratives, are largely factual or phyics related documents, with the books most represented by the topic being authored by Richard Feynmann.

Whilst not the focus of this investigation, is not necessarily on the topic models of narratives, it is interesting to see how topic models can represent differences in an authors writing. It suggests that they are are distinct differences between some authors, whilst there remain major similarities between others.

One concern is that in large series of narratives entities are often present throughout each document, perhaps resulting in any given topic's word distribution consisting primarily of entities from a series. When looking closer at the terms that form the topics, there are some topics that almost entirely consist of entities, whilst others are general terms relating to a specific subject.

The top terms for topic 8 are: 'Harry', 'Ron', 'Hermione', 'Percy', and 'Annabeth'. With 'Harry' being scored as twice as representative as 'Ron', labelled as the second most prominent entity. When considering the makeup of Topic 8, it is then clear why the Harry Potter novels feature such a high proportion of the topic.

The top terms for topic 0 are: 'Energy', 'Field', 'Example', 'Species', and 'data'. When looking at these terms, it is unsuprising to see that non-fictional and scientific documents recieve a high proportion of the topic.

Agatha Christie novels are strongly represented by topic 6, the top terms of which are: 'Poirot', 'Mr', 'Miss', 'Mrs', and 'Sir'. When looking in one Agatha Christie novel, 'The Mysterious Affair at Styles' There are 419, 180, 125, 234, and 113 mentions of each term respectively.

The primary take away from this figure, is that the topic modelling applied to the whole corpus has been able to produce groupings of related documents without any information other than

the terms within them. Whilst this is not a new finding when considering other literature, it serves as indication that the use of novels as a basis for corpora does allow topic modelling methods and to perform effectively, As the book level topic distributions across the corpus are somewhat similar, there is potential for entity topic models to vary for reasons aside from the novel they are in..

5.3.2 Harry Potter Summary Corpus Statistics

Formed as a corpus to evaluate the consistency of proposed methods, this corpus is formed as summary documents of each of the Harry Potter Novels by J. K. Rowling. The summaries are scraped from free online literature summary site 'SparkNotes' which compiles summaries of documents written by university graduates specialising in the subject being summarised. [55]

The corpus consists of 7 documents, with a total of 11,621 words ($\bar{x} = 1660$) and a total of 317 entities ($\bar{x} = 45$. The book level topic distributions are shown in , whilst this corpus is not expect to provide as many interesting insights into any hidden relations between entities of a narrative, it can serve as a means of evaluating elements seen as significant in the corpus of full Harry Potter Novels.
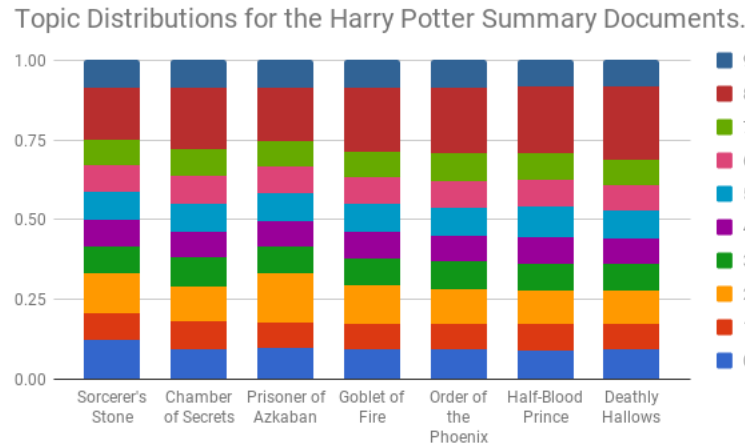
Figure 26: Topic Distributions for Summaries of Each Harry Potter Novel.

Books	Topic (%)									
	0	1	2	3	4	5	6	7	8	9
Sorcerer's Stone	12.38	8.35	12.56	8.39	8.18	8.70	8.43	8.08	16.42	8.51
Chamber of Secrets	9.54	8.52	10.80	9.25	8.27	8.64	8.89	8.30	19.34	8.44
Prisoner of Azkaban	9.60	8.03	15.71	8.35	8.06	8.50	8.48	8.02	16.77	8.48
Goblet of Fire	9.27	8.15	11.94	8.58	8.20	8.78	8.31	8.23	20.14	8.41
Order of the Phoenix	9.35	8.15	10.84	8.51	8.23	8.75	8.47	8.43	20.69	8.60
Half-Blood Prince	9.17	8.15	10.51	8.42	8.25	9.75	8.41	8.23	20.79	8.33
Deathly Hallows	9.28	8.03	10.59	8.38	8.01	8.68	8.07	7.90	22.91	8.15
$\bar{x}$	9.80	8.20	11.85	8.55	8.17	8.83	8.44	8.17	19.58	8.42
σ	1.15	0.18	1.87	0.32	0.10	0.42	0.24	0.18	2.31	0.15

Table 17: Proportional Distributions of Topic Models for the Harry Potter Novel Summaries.

0	1	2	3	4	5	6	7	8	9
professor	turns	flying	known	locket	locket	stone	outside	stone	elf
dog	winning	including	information	teaching	wand	sleep	forbidden	face	boys
dementors	realizes	wands	meet	way	night	following	takes	later	girl
night	ability	says	news	tower	friends	takes	day	tells	giant
large	soul	demanding	members	mr	mr	frustrated	announces	sees	left
later	magical	triwizard	teaches	named	eaters	hospital	informs	gets	cat
broomstick	mysterious	giant	knows	room	killed	moving	hoping	train	awful
game	away	padfoot	forest	wand	horcrux	brings	arrives	wizardry	night
parents	attack	divided	wakes	horcruxes	order	bathroom	godfather	dog	killed
time	game	voice	heart	members	new	street	corridor	named	return

Table 18: Top Ten Words in the Topic-Word Distributions of Calculated Topics for the Harry Potter Summary Corpus

5.4 Gaussian Entity-Term Matrix

Whilst the Gaussian Entity-Term Matrix (GETM) was defined only as a means overcome particular prior assumptions in literature it is necessary to evaluate the quality of associations formed within the matrix. Should the quality of underlying associations be poor, then the Entity-Topic models and Latent Entity Models will never be able to accurately capture entity structures within narratives. Whilst the reason for evaluating the GETM is similar to the rational behind evaluating the POS tagger used, the method for doing so is very different.

Without a pre-existing model or any testing dataset, evaluating the GETM can be done by considering the impact that using the GETM has on the performance of Entity Topic Models, or Latent Entities. Additionally, the consistency of the model across similar corpora can give insight into potential benefits and drawbacks of the method.

A consistency Metric was defined in chapter 3 for comparing two GETM calculated on different corpora, 3 different corpus pairs were used to evaluate the consistency of the method. The corpora in the first pair consist of books that are completely unrelated to books in the opposing corpus, the intuition behind this is to allow the method to have a baseline metric which the other corpora pairs can be compared to. The second pairing is constructed by splitting a novel series in half to form each corpus in the pair, whilst there will be distinct differences in the entities present across the documents, many of the entities in a series remain the same throughout meaning that there should be some consistency seen across the GETM models. The final pairing consists of factual documents on the same topics, the consistency of the models trained on documents should result in some consistency. Table ?? shows the corpora used in evaluating the consistency of the GETM model.

5.4.1 Harry Potter Corpus GETM
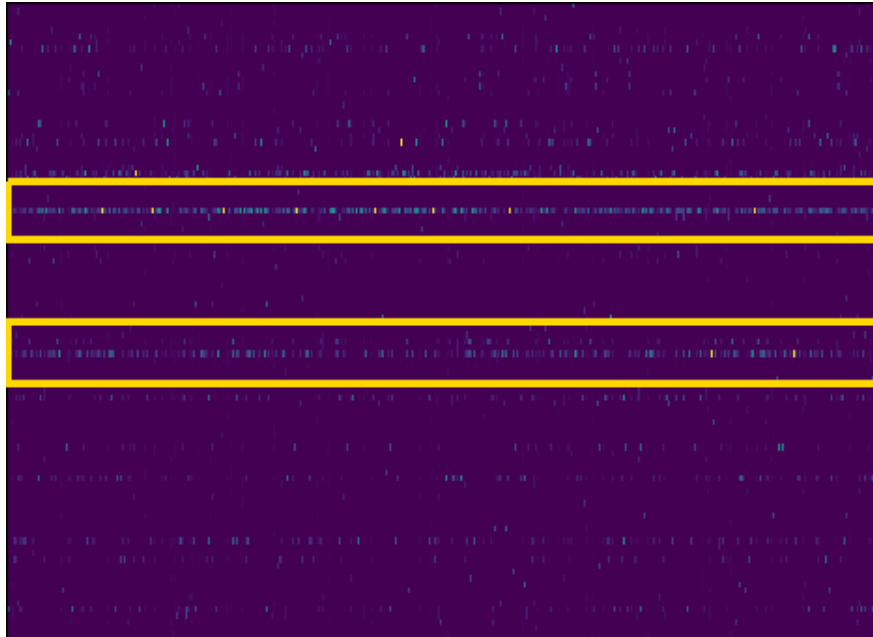
Figure 27: Portion of the Gaussian Entity Term Matrix for the full document of Harry Potter and the Philosopher's Stone

Displaying the GETM for a novel in a heat map form as shown in figure 27 really highlights the sparsity of associations between entities and terms. Only a select few entities really have some form of association with most terms throughout the corpus, with the majority of entities only having some degree of association with snippets of terms. When viewed in the context of a novel, this is a seemingly obvious observation, characters are frequently only present for specific scenes and portions of a book. The Harry Potter novels distinctly follow this structure, with entities like 'Katie Bell' being present in numerous books of the series, but mainly occurs in scenes oriented around the fictional sport of 'Quidditch'.

The significant Entities visible from the GETM are those forming light coloured bands across the darker purple, the most distinct bands are highlighted for clarity by the yellow boxes. The highlighted bands for this portion of the GETM in figure 27 are 'Harry', and 'Ron'. In the full GETM, entities 'Hermione' and 'Hagrid' are also significant bands. Notably, each highlighted entity characters of great significance to the story.

With respect to the rationale behind development of the method, the GETM does successfully

apply a limit to the degree of a association between entities and terms. Further exploration is required to evaluate the effects of the GETM on the calculation of entity topic models and latent entities.

The most significant entities in the philosophers stone according to the GETM are showed together in figure 28. The Term Matrix also encodes the significance that a term has to one entity in relation to its significance to other entities. Whilst there are light bands that show association between entities and terms, there are only a limited number of terms that are significant to that entity whilst being insignificant to most others.

Scaling from purple to yellow, as a term reaches the yellow end of the spectrum, the stronger the association there is between the entity and the term, in order for terms to reach the yellow end of the spectrum they must be associated strongly with an entity, whilst not being associated with other entities.

The top band in figure 28 shows the entity-term associations or the entity of 'Harry', some of the most significant terms include 'bloodcurdling', 'interesting-looking', and 'unbroken'. The consequences of these term associations would be seen when deriving Entity Topic Models, when calculating the topic models any word distributions for topics that place emphasis on these terms will form larger proportions of the Entity Topic Model for that entity, this will then have a knock on effect when calculating Latent Entities.
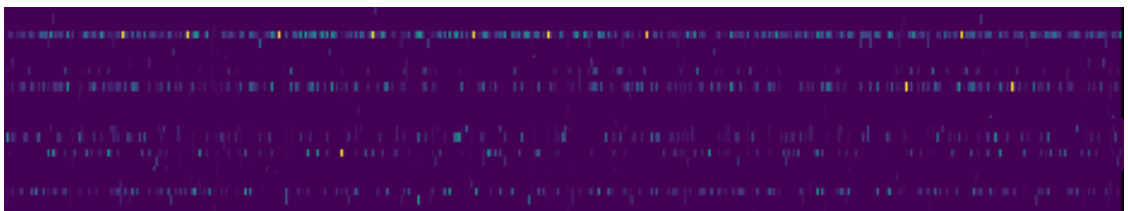
Figure 28: Significant Entities seen in the GETM for Harry Potter and the Philosopher's Stone.

5.4.2 Comparison To Summary Corpus GETM.
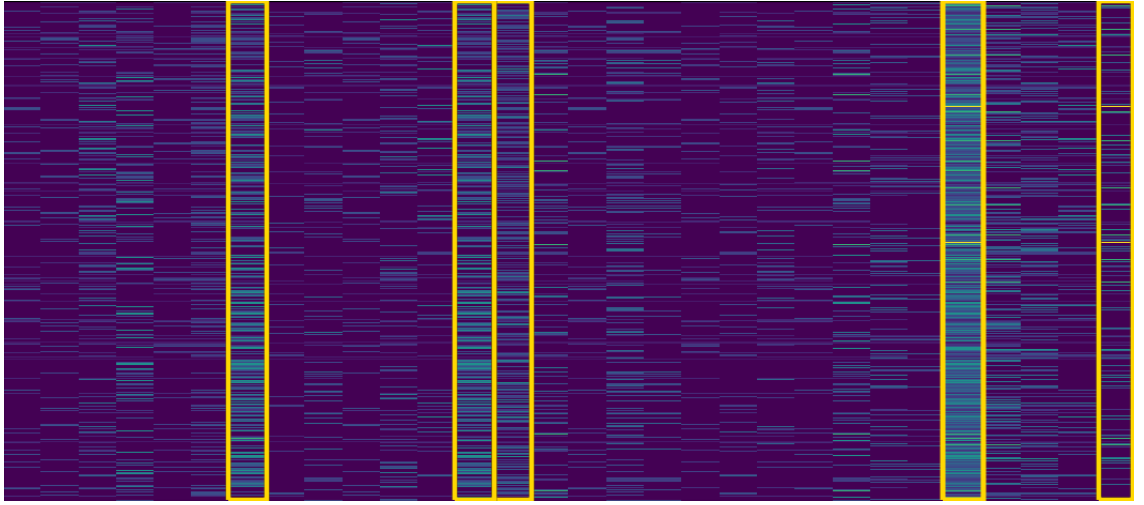
Figure 29: Gaussian Entity-Term Matrix for the Summary document of Harry Potter and the Philosopher's Stone.
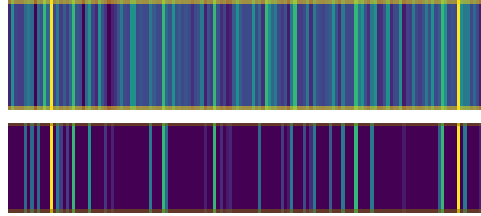
Figure 30: Magnification of GETM for 'Harry' and 'Quirrel' in the Summary of Harry Potter and the Philosopher's Stone

The corpus of Harry Potter summaries results in GETMs that are visually less sparse, each entity within it has a set of terms which they are associated with, with extensive overlap between many of the entities. Across each GETMS for the summary corpus, the major entities are 'Hermione', 'Ron', and 'Harry', which are highlighted by the first 3 yellow box from the left to the right of figure 29, these entities have strong associations with all with almost all terms in the document as well as with terms accross the corpus, for this GETM describing the summary document for Harry Potter and the Philosophers Stone, Hagrid is also a significant Entity.

A notable quirk of the model is that the entities 'Quirrel' and 'Harry' strongly associate to common terms that other entities do not, perhaps suggesting an underlying relationship between those entities. Additionally, despite the relatively short length of the document there still remains terms that remain significant to some entities whilst remaining disassociated with others, this suggests that even accross shorter documents, assumptions made in literature regarding the relationships that entities have to terms are fundamentally flawed.

5.4.3 GETM Interpretation

The GETM visualisation could provide an additional utility through which narratives could be analysed. When applying the method to Alice's Adventures in Wonderland, associations between terms and entities produced some interesting results for interpreting narratives latent meanings in narratives.

Considering the novel Alice's Adventures in Wonderland, Figure 22 highlights the top terms associated with the entity 'Wonderland'. Whilst requiring human interpretation (and some degree of artistic license), the GETM could be interpreted as describing 'Wonderland' as being nothing more than a 'Believed wonderland' and is in reality a 'suppressed misery'. Whilst interpretations such as these are subjective and often personal reflections, it is notable to consider that it was a numerical device that assisted in the task.

5.5 Entity Topic-Models

The use of a GETM was proposed as a means removing an assumption made in literature regarding the relationships which entities have with the topics within a document. It is therefore necessary to evaluate how the adapted approach affects the production of ETMs.

Replicating the assumption than an entity is related to all topics in a document can be attained by substituting the GETM weighting function with a function returning a constant value. This constant value places equal equal importance to terms in a document, as well as equal associations between entities and terms.

The resulting structure of Entity Topic Models would then be representative of topic distributions for the whole document, which is an acceptable structure when considering inter-document structures alone, however it does not consider the influence of latent intra-document structures on the relations across the corpora.

Notably, the GETM has not resulted in entities entirely dissimilar to book-level topic distributions, but it does allow for variability between entities. The results of the entity topic modelling method implemented are explored in the corpus of full harry potter narratives as well as the summary documents.

5.5.1 Harry Potter Corpus ETMs
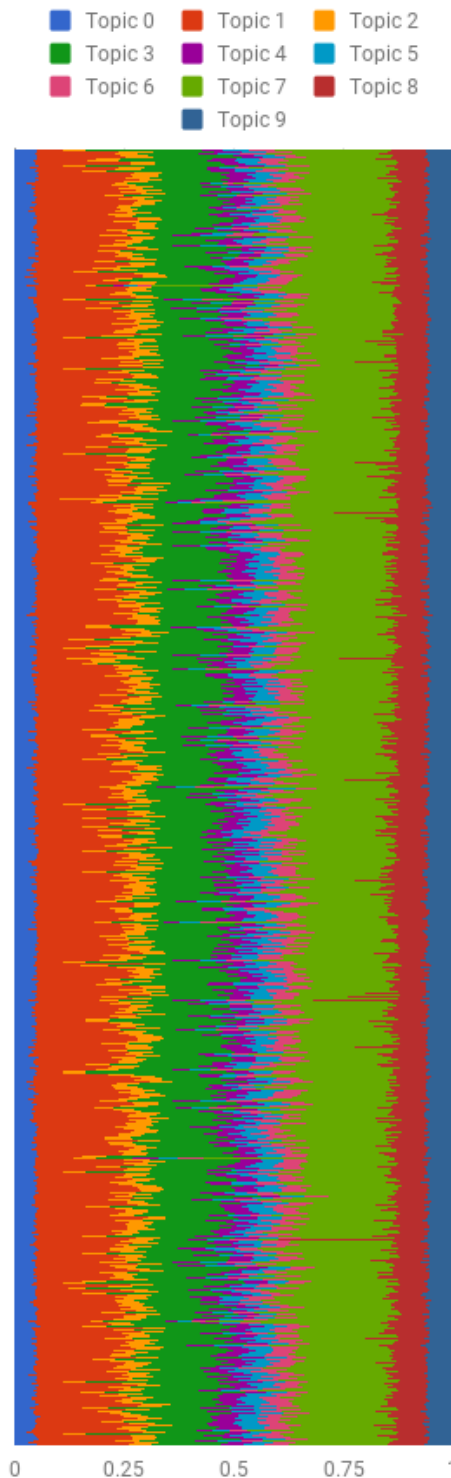
Figure 31: Book Topic Distributions for a Large Corpus of Fiction and Non-Fiction Books

The proportional distributions for all entity Topic models for the Corpus of Harry Potter novels are shown in figure 31. Entity Topic models are grouped according to the origins of the entity that the Entity Topic Model describes, distinct divisions are left out of the figure for clarity.

The general structure across Entity Topic Models is that topics 1, 3, and 7 are the most significant to all characters. This is somewhat different to the significant topics at a corpus level, which saw Topic 8 being one of the most significant. There are Entity Topic Models that are strongly represented by topic 8, however they do contrast the larger population.

As well as being significant proportions of all entity topic models, topics 1, 3, and 7 also see the largest amount of variation amongst, with topic 8 being the next most varying topic. By observing the variability of specific topics, some predictions can be made regarding which topics will influence the calculation of latent entities. When using K-Means clustering for calculating Latent Entities, it is likely that the topics of variance will form the basis for latent entities.
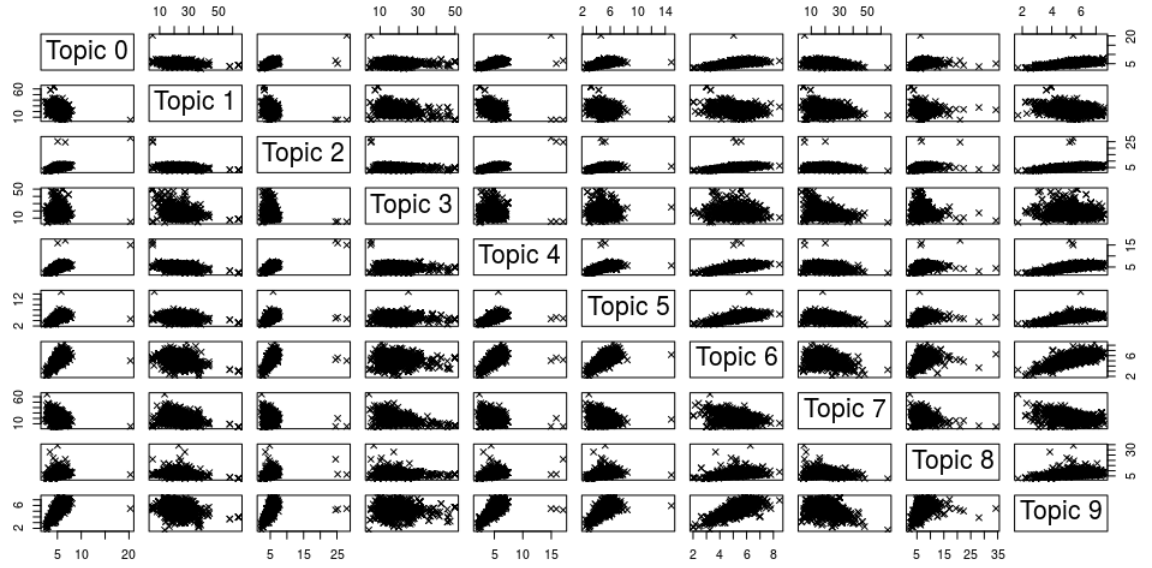
Figure 32: ETM Pairwise Scatter Matrix for Entities in the Harry Potter Novels.

A pairwise matrix scatter plot of entity topic model topic proportions can show some insight into the variance and distributions of topics across corpus of entities. Figure 32 confirms the significance of topics 1, 3, and 7 with regards to differentiating between entity topic models. The figure also highlights small but distinct groupings of entity topic models in Topics 0, 2, and 4, where the small groups consist of entity topic models feature significantly higher proportions of that topic when compared to the corpus at large.

For many of the topics, there are no obvious groupings of entity topic models, only what seems to be a linear distribution with a slight correlation between the topics plotted together. Topic 9 is one such topic that seems to have a proportional relationship with some topics. suggesting that for some entities, as the proportion of one topic increases, so does the proportion for topic 9. For this effect to take place it relies on the entity topic models being strongly composed of terms that are shared across multiple topic's word distributions, the shared terms however cannot be present across all word distributions for topics, otherwise the proportional topic distributions for an entity topic model would not change. The significance of this is that it suggests the entity topic modelling approach taken is capable of capturing some of the underlying semantics of words, which aligns with claims made in literature that certain modelling methods like latent semantic analysis are capable of doing such tasks.
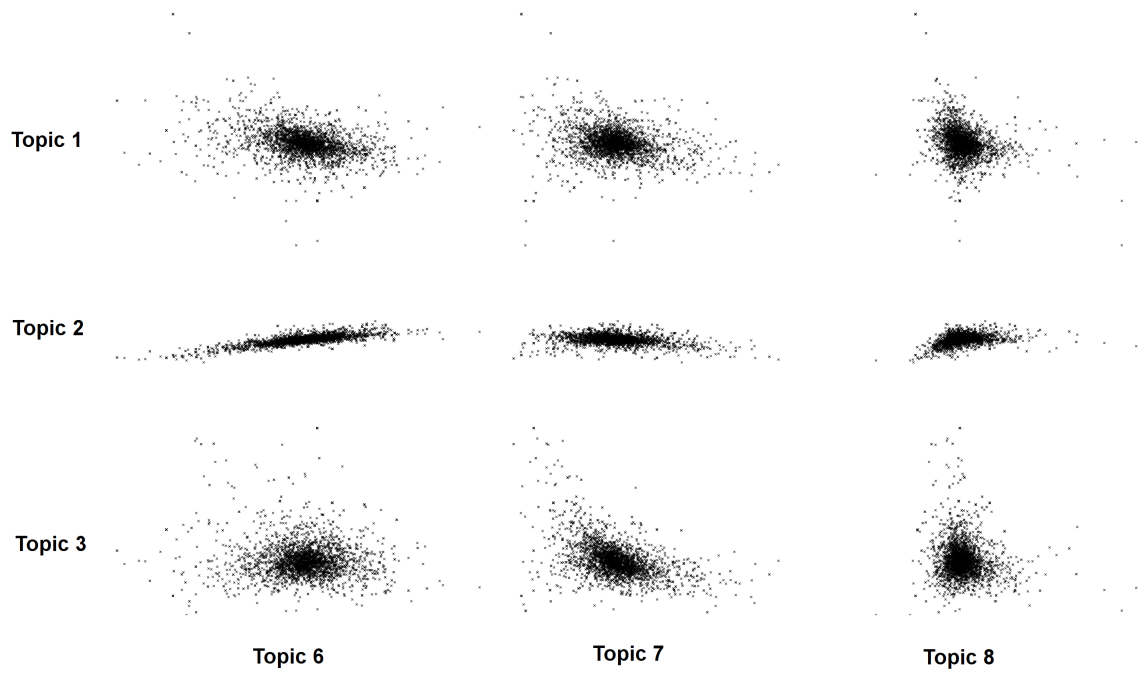
Figure 33: ETM Pairwise Scatter Matrix Zoomed in on A Selection of Topics of the Harry Potter Novels.

The relationships between topics across entity topic models can be seen a little more clear when magnified. Figure 33 shows a closer look at pairwise scatter charts for topics 1, 2, and 3 against topics 6, 7, and 8. For topic 2, there is a positive correlation between topics 2 and 7 whilst there is a negative correlation between topic 2 and 7. When considering the ability of topic modelling methods abilities to capture some of the semantics of words, then the correlations would be explained if the topics were formed of opposing or oxymoronic words, or if the topics are formed of terms that grammatically would not make sense if used close proximity.

Some subtle groupings can be identified in topics 3 against 7, in which there is a solid core of entity topic models, with 3 branches reaching roughly north, east and south west of the central core. This suggests that some of the relationships that may exist between terms, topics and entity topic models may be dependant on the entity involved.

5.5.2 Summary Corpus ETMs

For the corpus of full Harry Potter Novels, the Topic proportions for each Entity Topic Model roughly reflected the book level topic distributions. As a significantly shorter corpus, the summary documents see even less deviation from the book level topic distributions. The key difference between the Entity Topic models and the book level topic distributions is which topics are significant in distinguishing between elements. The summaries varied the most between topics 2 and 8, however the entity topic models see wider variance between topics 4, and 5.

When considering the Entity Topic Models for the full novel corpus, it was also observed that the topics of difference between novels were not the same as the topics of difference between latent entities. It was originally assumed that the assumptions made in literature regarding entities and their relations to topics in a document were valid for smaller documents, however this consistency between the small and large corpus suggests that there is perhaps an argument to be made that irrespective of size, entities may have different degrees of association with different elements of a document.

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
$\bar{x}$	8.44	9.75	8.68	8.44	8.47	8.96	18.38	8.74	8.64	11.50
σ	0.25	0.71	0.32	0.22	0.25	0.35	1.66	0.24	0.22	1.24

Table 19: Average and Variance of Topic Proportions across Entity Topic Models in the Summary Corpus
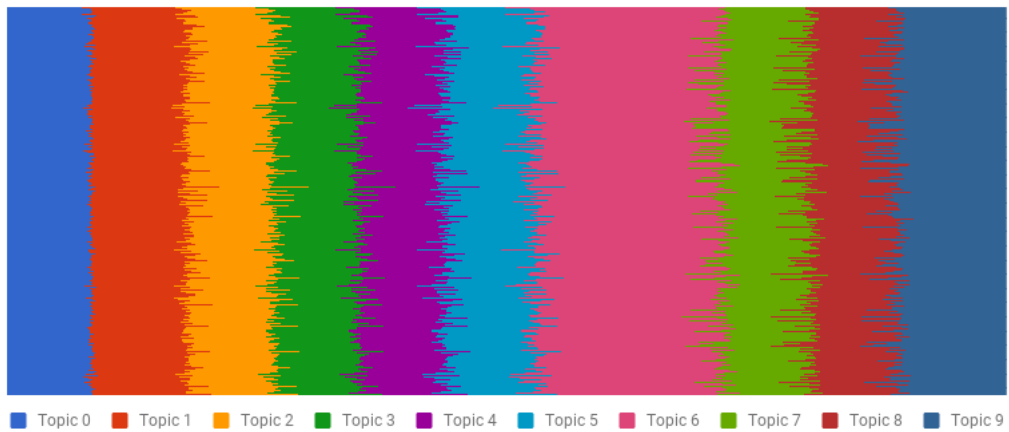
Figure 34: Proportions of Topics Forming Entity Topic Models within the Summary Corpus
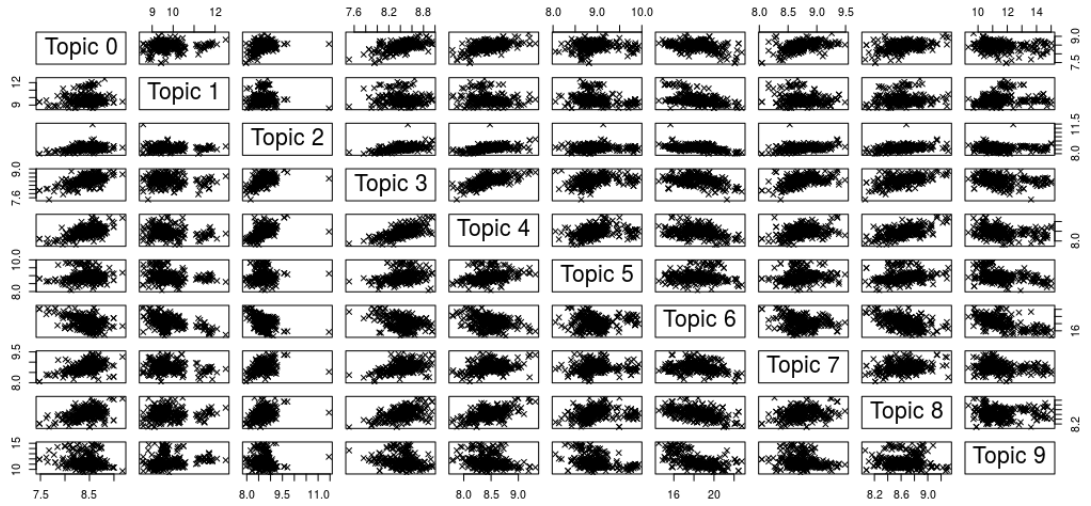
Figure 35: ETM Pairwise Scatter Matrix for Entities in the Harry Potter Novel Summaries.

In figure 34 you can see distinct spikes across some entity topic models in topic 9, topic 1 and to some extent, topic 6. Some of these spikes can be seen in the form of distinct groupings of entities within figure 34. The pairwise scatter plots for topic 1 and other topics show a completely separated group of entity topic models, and whilst not entirely separated, topic 9 can also be seen to have two groups, particularly when topic 9 is compared to topic 8 or topic 1. These groups could suggest a potential classification that can be applied to Entity Topic models in the form of a Latent Entity.

When you compare the pairwise plot of the Entity Topic Model Topics for the summary corpus, to the plot for the full corpus you can see that there are some similar structures formed. For the full corpus, there were topics that had a correlated relationship, where for some topic pairs, as the proportion of one topic in an entity topic model increased, the proportion of the other topic would decrease. There were also topic pairs that had the a positive correlative relationship, such that as the proportion of one topic increased, so did the other. With these similar formations observed across both corpora it suggests that the ETM approach used is capable of capturing some of the semantic structures within a corpus, or that the two processed corpora are similar in contents, meaning that the ETM method used has captured some of the semantic similarities between the two corpora.

5.6 Latent Entities

The evaluation of Latent Entities is a challenging task for two reasons. The first challenge is the lack of labelled data sets which a models performance can be evaluated on, the second difficulty is the subjective nature of the context for which the methods have been developed. Given meanings and interpretations of narratives and literary works are frequently debated and discussed within an academic setting, with interpretations varying from individual to individual, it raises the question of whether there ever could be a data set through which Latent Entity Models can be effectively evaluated.

There have been two metrics for Latent Entities that this thesis proposed as a means of evaluating the quality and value of produced models. Consistency is a proposed metric for evaluating the similarity of models trained on different corpora, and is of value for both Latent Entity models, as well as the previously discussed GETM. The second metric is a measure how well an individual can interpret a model which indicates how cohesive a latent entity model is. With no scope in the project for conducting meaningful surveys with a large sample size, a more general evaluation and interpretation of the topic models will be provided.

The full set of Harry Potter Novels will be explored before moving onto the corpus of summary documents. Exploration of the summary documents will be carried out by comparing and contrasting the calculated models to the full corpus of novels. The comparison aims to provide insight into any strengths and limitations that may be observed.

5.6.1 Harry Potter Corpus Latent Entities

To shows some visual insight into potential entity classifications entity topic models were plotted in a pairwise matrix of scatter charts, where each element in the matrix is the values of one topic against another, and colouring each entity topic model according to the extracted latent entity. Figure 36 shows the make up of five latent entity classifications that were extracted from Harry Potter corpus.

As with the book level topic distributions, there are some topics that offer little variability between entities. Topic 0 appears to provide minimal insight irrespective of the other topics. In contrast, topic 7 indicates a distinct divide between entities, however this also appears to be irrespective of other topics.

It is important to notice the different scales that are used across the pairwise plots of figure 36 Whilst topic 6 indicates a wide spread of entities, the range is actually more condensed than the ranges of other topics. should the scale for topic 6 be applied to seemingly dense topic plots seen for topics 0, 2, and others, and perhaps it might be visually more clear how the latent entity clusters are formed in those topics.

In all plots that show a tight grouping of entity topic models, there are also extreme outliers. These entities have not been clustered as a distinct classification in their own right, and perhaps raise a question as to whether they should be, however there is no consistent classification across the outliers for each pairwise plot, which suggests that the entity topic models forming outliers in some topics, are more closely related to other entities in other topics.

There is a seeming threshold for some topics that would result in an entity topic model being classified as a latent entity, for example, topic 1 sees all entities represented strongly by that topic classed as the green latent entity, topic 3 is the same but with the latent entity shown by black crosses, and topic 7 with the red coloured latent entity, this further suggests that certain topics play no role in forming a class of entity topic model.
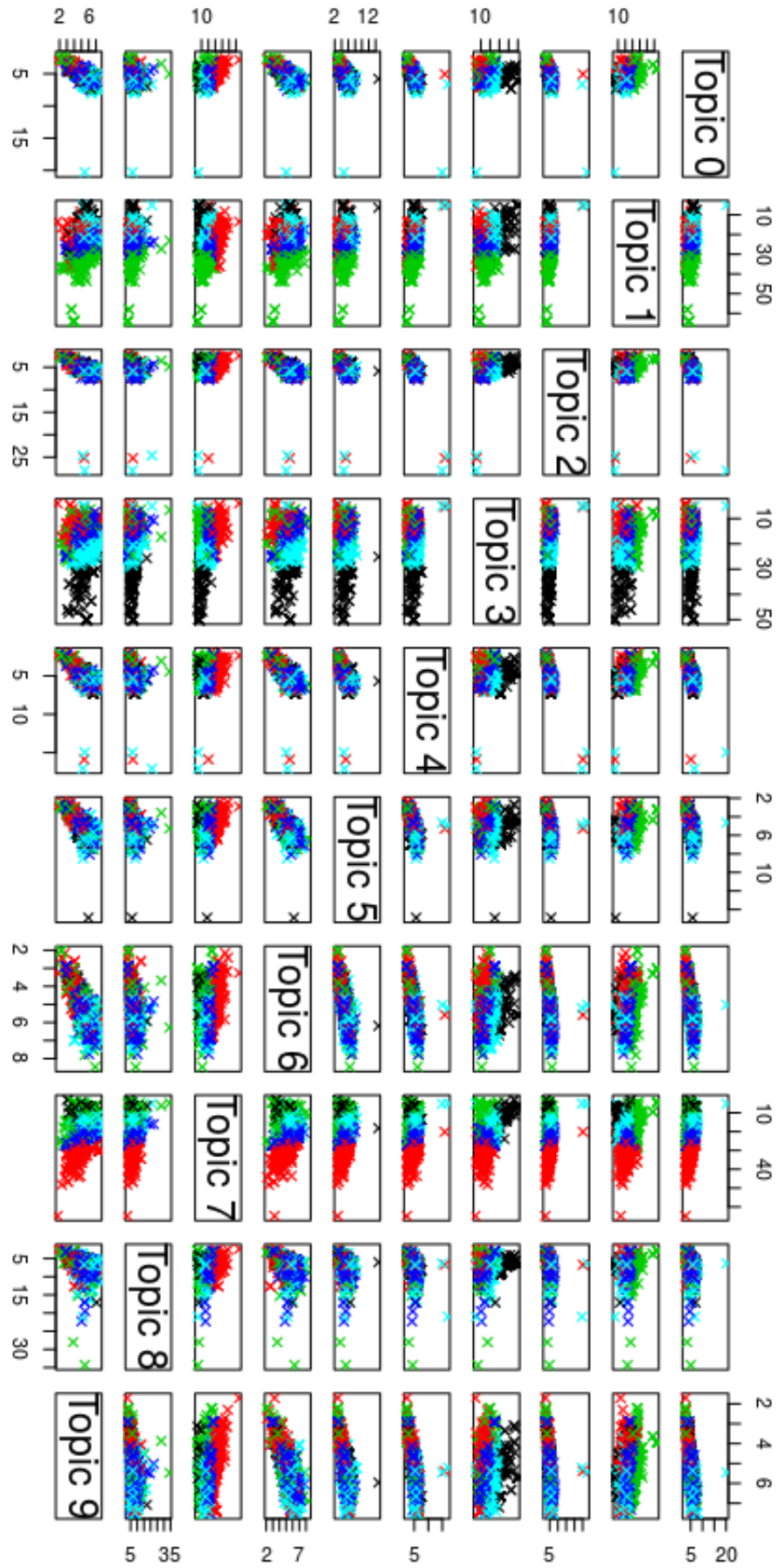
Figure 36: Pairwise Scatter Matrix of Entity Topic Model Topic Distributions Coloured According to Latent Entity they are Most Closely Associated With.

The density of the plots in figure 36 makes it difficult to see the significance of some classes within each plot. Expanding the view for a few topics, figure 37 shows more insight into pair wise plots of topics 6,7, and 8 against topics 2, 3, and 4. When viewed this way an estimate of the dominant classifications can be viewed a little more clearly. Across the plots for topic 3 (bottom row) the significance of some of the latent entities can be seen a little more accurately, the latent entity marked by black points is formed by such a small proportion of entities when compared to the latent entity marked by blue.

Topic 3 when inspected on a larger scale shows a much more varied spread of the entities that originally appeared to form a distinct classification, however some structure can still be seen when comparing distributions of topic 3 to topics 7, and 8. (bottom-middle, and bottom-right).

The magnification also indicates the dominant latent entity the corpus a little more clearly, the latent entity indicated by blue is an approximation for far more entity topic models than the other potential classifications.

Figure 37: Magnification of Pairwise Scatter Matrix of Entity Topic Model Topic Distributions for Topics 1,2 and 3, against topics 6, 7 and 8.

5.6.1.1 Latent Entity Composition.

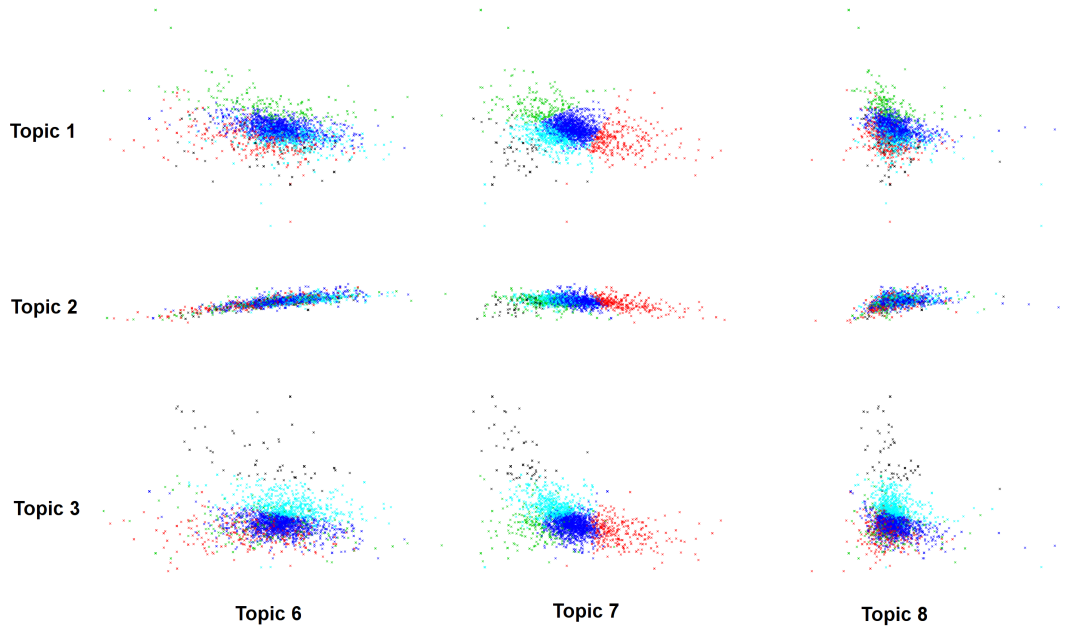| Latent Entity 0				Latent Entity 1				Latent Entity 2		
Entity	Distance	Origins		Entity	Distance	Origins		Entity	Distance	Origins
gorgovitch	2.67	Deathly Hallows		theodore	1.44	Order of the Phoenix		auror	1.51	Half-Blood Prince
georgie	3.90	Deathly Hallows		parvati	1.76	Half-Blood Prince		savage	1.51	Half-Blood Prince
miranda	4.14	Goblet of Fire		johnson	1.86	Prisoner of Azkaban		proudfoot	1.51	Half-Blood Prince
goshawk	4.14	Goblet of Fire		cho	2.05	Prisoner of Azkaban		blinky	1.61	Goblet of Fire
levine	5.09	Philosopher's Stone		maxime	2.14	Order of the Phoenix		krum	1.74	Deathly Hallows

Latent Entity 3			Latent Entity 4		
Entity	Distance	Origins	Entity	Distance	Origins
katie	0.39	Half-BloodPrince	beedle	0.70	Deathly Hallows
malfoy	0.46	Goblet of Fire	flamel	0.91	Sorcerer's Stone
pansy	0.56	Goblet of Fire	hagrid	1.01	Sorcerer's Stone
uncle	0.61	Goblet of Fire	petunia	1.14	Half-Blood Prince
cooperation	0.61	Goblet of Fire	bellatrix	1.17	Deathly Hallows

Table 20: Entity Topic Models most strongly represented by each Entity Topic Model.

Entities are deemed most closely associated, or best represented by a latent entity by calculating the euclidean distance between the topics of an Entity Topic model and the topics of a Latent Entity. The shorter the distance, the better representation the latent entity is for that entity topic model.

For the full corpus of Harry Potter Novels, the five entities best represented by each Latent Entity is shown in table 20. Notably, there is a diverse mix of characters in each latent entity with regards to the narrative which they are from, with four out of the five Latent Entities' top five entities being composed of entity topic models from three different narratives. When looking at more entity topic models for each latent entity, the diversity continues at the same rate.

There is only one novel not contributing to the top 5 entity topic models for any of the latent entities, however entity topic models from the Chamber of Secrets are represented in the top 150 entities for each latent entity. This could have implications regarding the nature of characters within the novels, either from the descriptions of them, or the context which they are found, entities in the Chamber of Secrets might distinctly different to the rest of the series. Alternatively it could be a result of the process calculating the latent entities, there could be a need for additional clusters so that entity classifications can be better represented. It should be noted however, that entity topic models forming the top 150 entities for a latent entity form only 4.5% of all entities

5.6.2 Summary Corpus Latent Entities.

Despite having fewer numbers in the corpus, the same number of latent entities were extracted. Figure 38 shows the topic proportions of each Entity Topic Model, with each point on figure coloured according to which latent entity the associated entity topic model was classified as.

Even though there are significantly less entities in the summary corpus when compared to the corpus of full novels, there is similar spreads of entity topic models in terms of the range of topic distributions seen in both the summary corpus and the full novel corpus, this is particularly true in the topics that have the most variation in them. What is even more striking is the similar patterns of clustering, there is still degree of similarity visually. Even though isn't an exact mapping between Latent Entity clusters, but there remains the impression that only a select number of topics are actually key to extracting Latent Entities.

For the full corpus, topics 1, 3 and 7 gave a clear indication of a threshold which was being used in grouping entity topic models, for the summary corpus, there is again 3 topics (1, 6 and 9) that provide a clear separation in groups of latent entities. Topic 1 shows a clear separation between the cyan latent entity class and the others, and topic 9 does the same for the green Latent Entity class. Topic 6 shows 3 distinct bands for red, blue, and black latent entity classes. Whilst the colours don't map to the full corpus, the same patterns occurred within its Latent Entity clustering.

Whilst visually promising, there is still further investigation required in order to evaluate the quality of the Latent Entity Modelling method. Consideration of the contents of each Latent Entity should be evaluated before conclusions can start to be drawn.

Given the sparsity of the pairwise plot for the summary corpus, a magnified view provides little insight.
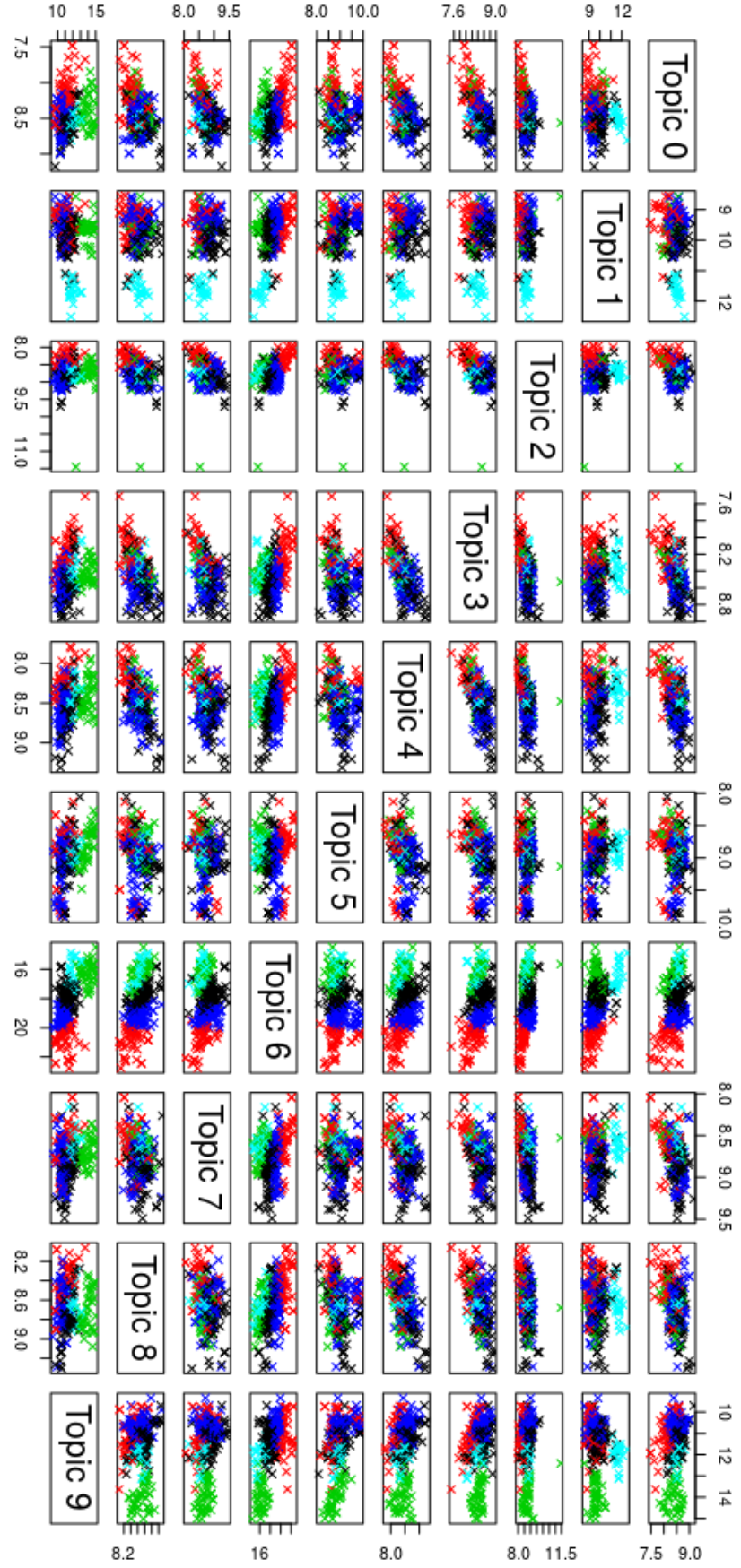
Figure 38: Pairwise Scatter Plot Matrix for Summary Corpus Entity Topic Models Coloured According to their Latent Entity Classification

5.6.2.1 Latent Entity Composition

Latent Entity 0				Latent Entity 1				Latent Entity 2		
Entity	Distance	Origins	Entity	Distance	Origins		Entity	Distance	Origins	
krum	0.46	Goblet of Fire	school	0.32	Order of the Phoenix		hagrid	0.32	Prisoner of Azkaban	
viktor	0.48	Goblet of Fire	witchcraft	0.39	Order of the Phoenix		voldemort	0.33	Prisoner of Azkaban	
cedric	0.48	Goblet of Fire	moody	0.62	Deathly Hallows		malfoy	0.53	Prisoner of Azkaban	
hagrid	0.54	Goblet of Fire	nagini	0.70	Deathly Hallows		mcgonagall	0.70	Goblet of Fire	
weasley	0.58	Chamber of Secrets	dolores	0.73	Order of the Phoenix		dumbledore	0.72	Prisoner of Azkaban	

Latent Entity 3			Latent Entity 4		
Entity	Distance	Origins	Entity	Distance	Origins
hermione	0.26	Order of the Phoenix	hagrid	0.48	Philosopher's Stone
harry	0.34	Order of the Phoenix	harry	0.55	Philosopher's Stone
neville	0.43	Deathly Hallows	dursley	0.57	Philosopher's Stone
sirius	0.43	Order of the Phoenix	voldemort	0.61	Philosopher's Stone
voldemort	0.49	Order of the Phoenix	hermione	0.62	Philosopher's Stone

Table 21: Top 5 Entity Topic Models Forming the Latent Entities for the Harry Potter Summary Corpus.

Considering the reduced set of Entity Topic Models that Latent Entities can be calculated from, it is of no surprise to see entirely different entities forming the core of each Latent Entity. There are a few similarities between the sets of latent entities, particularly when he origins of the entity are ignored. The entities 'krum', 'malfoy' and 'hagrid' are common to both sets of latent entities. Whilst there are multiple 'hagrid' entities in the top 5, the occurrence of note is the 'hagrid' originating from the 'Philosopher's Stone', which matches the occurrence in the Latent Entities Extracted from the full novel corpus.

5.6.2.2 Consistency

Cherry picking entities is not necessarily the best way of evaluating the similarities of the the two sets of latent entities. As a k-means clustering approach was taken, it is possible re-use the classifier of the summary set of entities on the entity topic models from the full corpus. By measuring the numbers of entities classified together from both classifiers, you can get some estimate as to whether the latent entities extracted from either corpus are the same. Additionally, using the previously defined consistency metric can provide more insight into the similarities of both sets of Latent Entities.

The consistency metric defined previously involved treating one set of latent entity compositions as being the *de facto* set of entities, and then checking if each entity pairing in the other corpus

is present in the *de facto* corpus.

With matching entity topic model pairings being counted as true positive (TP) and non matching pairings being counted as false positives (FP) the consistency metric is defined as:

$$Consistency = \frac{TP - FP}{\sum_{i=1}^{N} \binom{n_i}{2}}$$

Where N is the number of Latent entities clusters, and n_i is the number of entities within the latent entity cluster indexed by i.

As the number of entities is drastically different between the two corpora used in this thesis, only entities present in both will be considered. Only using common entities ensures that the the metric has an emphasis on similarities as opposed to the differences.

The consistency scores for the Latent Entities extracted from the full corpus and the summary corpus are as follows:

$$TP = 3,815, \qquad FP = 8,549, \qquad \sum_{i=1}^{N} \binom{n_i}{2} = 12,364$$

$$consistency = -0.38$$

Whilst this score is negative, implying that the Latent Entities are largely inconsistent between the two corpora, a score greater than -0.5 means that there is still a significant amount of consistent entity topic models being represented by the same latent entities.

It should be noted that this metric punishes inconsistencies greatly, a single difference across a latent entity will result in the FP being increased by the size of the cluster the entity was misplaced in. Additionally, if a single latent entity encompassed all entity topic models present in an opposing corpus, then a perfect consistency score would be given, even though there is really no consistency.

5.6.3 Cohesion and Interpretation

To effectively evaluate the cohesion and interpretation using the proposed intruding entity metric requires a large sample of participants as well as an extensive suite of questions is required. Without a large sample, the validity and reliability of any conclusions is always going

to be in doubt. It was determined early in development that attempting to survey a large sample would be entirely out of scope for the project, and as such accurately evaluating the cohesion and interpretation of the models is difficult to do.

In order to provide some estimate of the cohesion and interpretability of the models, a personal evaluation on the models is given, it is acknowledged that this is not sufficient for real conclusions to be drawn, but any insight could be used as a basis for further study.

5.7 Explorer Evaluation

Evaluating the implemented Explorer web application and associated system requires consideration of the rational behind why the system was required. The quality of the implemented system can be assessed a side from any strengths or weaknesses of the models that the system supports.

With regards to usability of the system, there are two aspects to the system, the command line interface, and the web application interface. The command line interface is simple and requires providing the directory of the corpus files which are to be analysed, there are no other settings for configuration. The limited interface somewhat reduces the flexibility of the analysis pipeline, built within the NLP toolkit is flexibility with regards to some of the hyper parameters within implemented models, including the choice of weighting function, as the number of topics, the number of terms to include within each topic, and the number of latent entities which the system should derive. As a result of excluding these parameters from command line arguments, the pipeline requires manually configuring from within the code base, reducing the accessibility and usability as a whole.

The Explorer Web Application however is a much more accessible tool when compared to the command line interface of the pipeline. It adheres to principals outlined in the Google Material Design specification providing bold and contrasting colours to provide a clear interface. The incompleteness of the interface is its largest drawback. For the interpretation of simple topic information the program is an effective window into the corpus. However the lack of simple visualisations reduce the scope of any investigation that an individual wishes conduct through the interface.

There are some aspects of accessibility where the system is somewhat effective. Primarily it is in the way the system segments data about the corpus into logical sections (Book Data, Topic Data, and Entity Data). By segmenting information in this way, there is less risk of users being overloaded with numerical data. There is still room for improvement however, instead of

presenting a panel of numerical comparisons of entity topic models, a dynamic figure could be implemented that allows for visual and interactive comparisons to be conducted.

Considering the amount of data that can be extracted from a large corpus, the each request for lists of Books, entities, and topic information took on average $455ms$ for the application to receive after issuing the request. For a general indication as to whether these times are good or bad, Nottingham Trent University's Online Workspace (NOW), took over 10 seconds to load aspects of the page ($\bar{x} = 7.42 seconds, \sigma = 0.72 seconds$). The significantly longer loading times NOW suggest that the loading times of the Explorer application are somewhat acceptable.

It should be noted that the assessment of load times is only against one website that has numerous users and transfers images as well as text information. The comparison is only to state that if 10 seconds of waiting for resources is acceptable, then 455ms is also acceptable.

Chapter 6

Conclusion

6.1 Introduction

This thesis has presented, explored, and evaluated the concept of Latent Entities, a form of Topic Modelling that sits as a layer of abstraction on top of a collection of Entity Topic Models. The calculation of a set of Latent Entities for a corpus depends on numerous pre-processing steps including the extraction of entities and the calculation of Entity Topic Models. Classical approaches to Entity Topic Modelling fail to take into consideration the associations that entities within a narrative has with other terms in a document, and as such this thesis also explored the possibility of using an adapted co-occurrence matrix as a means of incorporating these features of narratives in the calculations of Entity Topic Models.

Given the abstract nature of topic modelling methods, alternative methods of evaluation have been considered. Visually exploring the results from two sets of related corpora provided some insight into the value of the methods as well as their shortcomings. A consistency metric was proposed and applied two related corpora to investigate the consistency of the resulting latent entities.

To facilitate the exploration of Latent Entities a small system was implemented, the system was composed of an analysis pipeline, server architecture, and web application. Whilst only a secondary objective, the system was essential as a means of ensuring robust software methods for analysis were implemented.

To support the exploration of Latent Entities the results from numerous other elements of the implemented analysis pipeline were explored, including the associated set of Entity Topic

Models, sets of Gaussian Entity-Term Matrices, and book level Topic Models.

6.2 Outcomes

This thesis was successful in meeting its aims and objectives. An analysis pipeline, exploration interface, and consideration of novel methods of analysis were investigated and implemented. With regards to tangible outcomes, this thesis has now made available an Open Source suite of tools that others can use to explore and investigate Latent Entity Modelling.

The implemented pipeline provides an extensive and robust code base that is ideal for future investigatory work. Despite the completeness of the suite, there are still drawbacks with regards to deployment as well as the interface through which the pipeline can be used. There are no tangible means of deploying the pipeline besides downloading and including the code base inside whatever project is being worked upon.

A second tangible result of the thesis is an extensible API and Server system. By providing access to results from analysis, further investigatory and exploratory work on the Latent Entity Models can be carried out. As an Open Source implementation, it is easy for other developers to expand the API should there be required aspects that are unimplemented. The primary downside to the system is similar to the implemented pipeline, with no real means of deployment included with the system, individuals wishing to utilise it for personal projects are required to manually set up and configure each distinct element, including configuration files and database administration.

The implemented interface is the weakest element of this thesis, it provides the minimum set of tools necessary for an individual to begin exploring the topical structure of entities and documents within a corpus. Whilst the visualisations maybe lacking in the explorer interface, the web application stayed within the bounds of the project scoping and satisfied the purpose for which it was developed.

An additional outcome involved the implementation of a Viterbi decoded Hidden Markov Model POS tagger. A valuable tool to have when undertaking NLP oriented projects, the tagger

performed somewhat better than expected. For closed classes of words the tagger performed comparably to taggers in literature, it was on open classes of words where hapax legomena were common that the implemented tagger had room for improvement.

6.2.1 Modelling Outcomes

As the focus of this thesis, the methods of analysis that were implemented are the strongest outcomes for the project, as three novel approaches to topic modelling were proposed, defined, and implemented.

The Gaussian Entity-Term Matrix provided a means of preserving and expressing entity-term relationships that may be present within a document. As an approach, it is an adaptation of a standard co-occurrence matrix and can be utilised in statistical modelling methods like Latent Semantic analysis. Exploration of the resulting matrices showed that the method was capable of identifying terms of significance to entities within a document, as well as highlighting entities that are of significance to a document.

An alternative approach to Entity Topic Modelling was then explored through the use of Latent Dirichlet Allocation and the Gaussian Entity-Term Matrix. The outcomes of the new approach to entity topic modelling is one that shows a potential to capture the semantic meanings and relationships of terms and entities within a document, as well as the semantic differences between terms and entities between documents in a corpus.

With an approach to Entity Topic Modelling that incorporated varying entity-term and entity-topic relationships across a corpus, Latent Entity Modelling was shown to be able to form approximate classifications for groups of Entity Topic Models. The exploration and comparison of two corpora showed that the approach was capable of consistently identifying structures formed by entity topic models across a pair of related corpora. The resulting weakness of the approach is seen in the consistency of entity classifications, the inter-Latent Entity consistency between corpora was weaker than was hoped.

6.3 Limitations

From the extent of exploration and evaluation of implemented models, to the limited breadth of visualisations implemented in the exploration web interface, there are numerous limitations to the work presented in this thesis. Consideration of implementation limitations are discussed first, followed by the limitations from the evaluation and exploration aspect of the thesis.

6.3.1 Implementation Limitations

Whilst the Web interface provides a starting point for encouraging individuals to explore and learn about data analysis, natural language processing, and topic modelling, nothing more than surface level understanding can be obtained in its current form. The interface relies on users being numerate and does not accommodate users that are not comfortable with reading and interpreting figures and listings. A key oversight for the interface is that there is no explanation of any of the components presented to the user, and no explanation of how the user interacts. Whilst some elements are clear, that cannot be said for the application as a whole. This ultimately limits the usability of the application to users already familiar with topic modelling as a field.

With no methods of deployment, easy means of configuration, and a requirement to manually configure large aspects of the pipeline, the reach that the system can have with regards to others using it for personal projects is severely limited. It also means that any expansion of the pipeline is a more challenging task. Whilst it is a successfully working prototype, there is a lot of refinement and formalisation that is required of the system before it could ever be considered a viable system for other researchers, businesses, or individuals to use.

As a method of preserving entity-term relationships, the Gaussian Entity-Term Matrix is somewhat successful, as a fast and efficient processing tool, the implemented algorithm is a drastic failure. A result of limited processing resources the model implementation requires entity data to be written to disk frequently, either to a file or to a database, the process of calculating the Gaussian Entity-Term Matrix for a large corpus can take weeks of processing.

As a result of pushing the results of the Gaussian Entity-Term Matrix to a database, the implementation of the proposed method of calculating Entity Topic Models relies on the database before any analysis can be done. As all results from the Gaussian Entity-Term matrix are required for the Entity Topic Models to be calculated, a hardware limitation is enforced. If there is not enough memory available in hardware, then it is impossible for this implementation to be completed. Whilst Topic Modelling approaches exist that allow for batches of processing to be carried out, this thesis does not implement or use any. Additionally the Entity Topic Modelling approach implemented requires the user to specify how many topics to build as well as how many terms are to form each topic, this could result in users having to run the full analysis pipeline numerous times before adequate parameters are found.

The Latent Entity Modelling method draws most of its limitations from the k-means algorithm. Similarly to the entity topic modelling method, calculating Latent Entities requires users to specify how many latent entities are to be calculated. The chosen number of latent entities can have drastic effects on the quality of resulting models. When considered in conjunction with the other parameters that users have to choose for themselves, the large number of possible configurations that a user must try before finding results is a severe limitation of this project.

6.3.2 Evaluation Limitations

Whilst this thesis explored a range of different methods of analysis, the depth to which each method was investigated is a large limitation. Each stage of the pipeline consists of configurable aspects that could feasibly alter the results of later stages of the pipeline. This project was limited in its investigation of what effect each configuration would have on the resulting methods, which limits the validity of any conclusions and insights that may have been gained.

Another key limitation of the project is the size and range of corpora that were used to evaluate the models. Whilst the corpora used provided a range of entities and did appear to be a valid option for analysis, without more corpora to compare the results to, there is insufficient evidence for any claims regarding each model's abilities to be validated. Using a range of corpora configurations, would provide more evidence towards the pipelines ability to identify

latent information and latent classes of entities.

Without a viable option for surveying a large population, the interpretability and thus usability of Latent Entity models cannot be effectively concluded. Whilst personal reflections are given on various interpretations of the latent classes that have been extracted, without knowing how other individuals interpret results then claims about their usefulness are somewhat invalid.

6.4 Future Work

Future work for this project is oriented around 3 key aspects, Exploration, Evaluation, and Interpretation. Whilst the thesis has demonstrated that Latent Entity Modelling may be a valuable addition to the field, it too many limitations that need to be addressed before the exact value of Latent Entity Modelling can be known.

As abstract representation of documents, interpreting and exploring topic models is not straightforward. It requires individuals to be both numerate and somewhat familiar with the field to really grasp the meaning to the models. There is potential for a project to focus around the visualisation of Latent Entities, with the aim of determining the most effective ways of communicating the meaning of the models as well as the most effective ways of enabling individuals to explore and interpret each Latent entity.

As a new concept, Latent Entities required new metrics for evaluation. Further investigation is required to determine exactly how Latent Entity Models should be evaluated. The proposed metric of consistency is not necessarily enough, especially given how much variation there can be between two corpora. The intruding entity test proposed in this project may be an effective way of evaluating Latent Entities and a full investigation of this would be interesting to see.

Not necessarily a research project, further development can be carried out that formalises and improves the quality of the implemented system and pipeline. The development of a wider range of visualisations for the prototype web application could provide a solid framework for an investigation into effective visualisations of Topic models and Latent Entity Models.

Bibliography

[1] bkkbrad and Wikipedia contributors. Latent dirichlet allocation. `https://en.wikipedia.org/w/index.php?title=Latent_Dirichlet_allocation&oldid=835569966`, April 2018. Accessed: 2018-4-15.

[2] Web server survey | netcraft. `https://news.netcraft.com/archives/category/web-server-survey/`. Accessed: 2017-10-16.

[3] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Comput. Linguist.*, 21(4):543–565, December 1995.

[4] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pages 112–116, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[5] H Huang and X Zhang. Part-of-speech tagger based on maximum entropy model. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 26–29, August 2009.

[6] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 133–140, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[7] Geoffrey Leech, Roger Garside, and Michael Bryant. CLAWS4: The tagging of the british national corpus. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 622–628, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[8] A Ratnaparkhi. A maximum entropy model for part-of-speech tagging. *of the conference on empirical methods in . . .*, 1996.

[9] G Malecha and I Smith. Maximum entropy part-of-speech tagging in NLTK. *unpublished course-related report*, 2010.

[10] Slav Petrov, Dipanjan Das, and Ryan T. McDonald. A universal part-of-speech tagset. *CoRR*, abs/1104.2086, 2011.

[11] Ewa Dąbrowska. What exactly is universal grammar, and has anyone seen it? *Front. Psychol.*, 6:852, June 2015.

[12] Roger Garside. *The computational analysis of English: A corpus-based approach*, volume 57. 1988.

[13] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

[14] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, 30(1):3–26, January 2007.

[15] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[16] Guodong Zhou and Jian Su. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 473–480, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[17] A Borthwick and R Grishman. A maximum entropy approach to named entity recognition. 1999.

[18] Oliver Bender, Franz Josef Och, and Hermann Ney. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 148–151, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[19] Erik Tjong Kim and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-Independent named entity recognition. June 2003.

[20] John D Lafferty, Andrew McCallum, and Fernando C N Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., June 2001.

[21] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[22] Samet Atdağ and Vincent Labatut. A comparison of named entity recognition tools applied to biographical texts. August 2013.

[23] J R Finkel, T Grenager, and C Manning. Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005.

[24] The stanford natural language processing group. `https://nlp.stanford.edu/projects/project-ner.shtml`. Accessed: 2018-1-10.

[25] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[26] Jimmy Lin. Scalable language processing algorithms for the masses: A case study in computing word co-occurrence matrices with MapReduce. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 419–428, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[27] Chris Biemann. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 73–80, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[28] Pradipta Maji, Malay K Kundu, and Bhabatosh Chanda. Second order fuzzy measure and weighted co-occurrence matrix for segmentation of brain MR images. *Fund. Inform.*, 88(1-2):161–176, 2008.

[29] Yutthana Munklang, Sansanee Auephanwiriyakul, and Nipon Theera-Umpon. A novel fuzzy co-occurrence matrix for texture feature extraction. In *Computational Science and Its Applications – ICCSA 2013*, pages 246–257. Springer Berlin Heidelberg, 2013.

[30] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(Jan):993–1022, 2003.

[31] Scott C Deerwester, Susan T Dumais, George W Furnas, Richard A Harshman, Thomas K Landauer, Karen E Lochbaum, and Lynn A Streeter. Computer information retrieval using latent semantic structure, June 1989.

[32] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Process.*, 25(2-3):259–284, January 1998.

[33] Cherukuri Aswani Kumar and Suripeddi Srinivas. On the performance of latent semantic indexing-based information retrieval. *CIT. Journal of Computing and Information Technology*, 17(3):259–264, October 2004.

[34] István Bíró, Jácint Szabó, and András A Benczúr. Latent dirichlet allocation in web spam filtering. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '08, pages 29–32, New York, NY, USA, 2008. ACM.

[35] Yansong Feng and Mirella Lapata. Topic models for image annotation and text illustration. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 831–839, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[36] Tong Zhang, Zhe-Ming Lu, Kap Luk Chan, and Zhen Li. Automatic image annotation and retrieval using the latent dirichlet allocation model. *IJCSES International Journal of Computer Sciences and Engineering Systems*, 5(1), 2011.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[38] Matthew D Hoffman. Online LDA.

[39] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[40] Michael A Newton and Adrian E Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *J. R. Stat. Soc. Series B Stat. Methodol.*, 56(1):3–48, 1994.

[41] Hanna Megan Wallach. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008.

[42] Siddhartha Chib. Marginal likelihood from the gibbs output. *J. Am. Stat. Assoc.*, 90(432):1313–1321, 1995.

[43] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.

[44] Pengtao Xie, Diyi Yang, and Eric Xing. Incorporating word correlation knowledge into topic modeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 725–734, 2015.

[45] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In Y Bengio, D Schuurmans, J D Lafferty, C K I Williams, and A Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009.

[46] Fred Morstatter and Huan Liu. A novel measure for coherence in statistical topic models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 543–548, 2016.

[47] Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22. aclweb.org, 2013.

[48] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[49] P J Crossno, A T Wilson, T M Shead, and D M Dunlavy. TopicView: Visually comparing topic models of text collections. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 936–943, November 2011.

[50] Xianpei Han and Le Sun. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 105–115, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[51] Paul Mcnamee and Hoa T Dang. Overview of the tac 2009 knowledge base population track, 01 2009.

[52] Yani Dzhurov, Iva Krasteva, and Sylvia Ilieva. Personal extreme Programming–An agile process for autonomous developers. January 2009.

[53] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, 32(4):485–525, November 2006.

[54] D Jurafsky and J H Martin. Speech and language processing. 2014.

[55] SparkNotes: Today's most popular study guides. `http://www.sparknotes.com/`. Accessed: 2018-4-21.