Title line 1
Title line 2

A thesis submitted in partial fulfilment

of the requirement for the degree of Doctor of Philosophy

Name M. Lastname

July 2011

Cardiff University
School of Computer Science & Informatics

Declaration

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

Statement 1

This thesis is being submitted in partial fulfillment of the requirements for the degree of PhD.

Signed .... (candidate)

Date ....

Statement 2

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed .... (candidate)

Date ....

Statement 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

A copy of this document in various transparent and opaque machine-readable formats and related software is available at `http://yourwebsite`.

To People you care
for their patience and support.

Abstract

We produce interpretable representations, and demonstrate their applicability in interpretable classifiers. Our approach is model-agnostic, given a similarity-based representation, we are able to produce a representation in terms of domain knowledge. We evaluate the interpretability of our representation and provide examples of interpretable classifiers with our representation.

Acknowledgements

Contents

List of Publications

The work introduced in this thesis is based on the following publications.

-
-

List of Figures

List of Tables

List of Algorithms

List of Acronyms

ML Machine Learning

NLP Natural Language Processing

NDCG Normalized Discounted Cumulative Gain

0.0.1 Definitions

Domain Where the data was originally sourced from DOM^IMDB, e.g. IMDB movie reviews.

Word A string of alphanumeric characters that originated from text in the domain DOM_w, e.g. the $w = "Horror"$ from a domain of IMDB movie reviews DOM^IMDB.

w

Corpus of Documents A unique group of words, e.g. a review from a domain of IMDB movie reviews DOM_IMDB.

C_dw

Document A document of words

d_w

Vector Space A representation composed of vectors.

S_v

Semantic Space A representation where spatial relationships between vectors correspond to semantic relationships.

S_v

Word frequency The frequency of a word wf for its document D_wf.

wf

Bag-Of-Words a matrix BOW of documents BOW_D where each document is composed of unordered frequencies of words $D = [wf_1, ..., wf_n]$. and Conceptual Space we obtain a representation of entities composed of properties. Then, we cover the additional methods we propose to improve this process.

BOW_d

Bag-Of-Words PPMI

Feature A feature is a distinct useful aspect of the domain, corresponding to a numerical value.

R_f

Hyper-plane The hyper-plane for a word

H_w

Direction vector The orthogonal direction to a hyper plane that separates a word in a vector space.

D_w

Cluster label A cluster of words that describe a property.

C_w

Cluster direction The averaged directions of all words in the label.

D_C

Feature rankings The rankings induced from a feature direction.

R_DC

Chapter 1

Introduction

1.1 Motivation

With the rise of services on the web that enable large-scale user-generation of text data, e.g. Social Media sites (Facebook, Twitter), Review sites (IMDB, Rotten Tomatoes, Amazon) and content-aggregation sites (Reddit, Tumblr), the internet has become largely populated by text posts that are related to some specific, niche topic within a domain. For example, a review on Amazon for a product is specially tailored text for that product within the domain of Amazon reviews. Taken from a closer lens, we could even argue that each review-type has its own domain, e.g. Product reviews, Food reviews, Movie reviews. However, the text posts themselves are largely unstructured semantically. Humans can have an intuitive understanding of the semantics that are present in unstructured text, but machines do not.

One task of Natural Language Processing is to obtain this semantic understanding from text by obtaining a machine-readable representation that contains domain knowledge. A basic approach to obtain a representation of this text is to represent entities (e.g. reviews, text-posts) by the frequency of their words, see 1.1.

Below, we show a review with its associated properties labelled.

We can understand these properties to have a degree to which they apply, for example the size of the clothing might be "XXL", "XL", "L", "M" or "S", or the quality may be "Very good", "Good", "Ok", "Bad" or "Very bad". For the former, we may rely

Entity: X

Word	Frequency
Dog	51
Cat	40
Man	11
Cheese	0
Dog	51
Cat	40
Man	11
Cheese	0

Entity: Y

Word	Frequency
Dog	51
Cat	40
Man	11
Cheese	0
Dog	51
Cat	40
Man	11
Cheese	0

Entity: Z

Word	Frequency
Dog	51
Cat	40
Man	11
Cheese	0
Dog	51
Cat	40
Man	11
Cheese	0

Figure 1.1: Bag-of-words

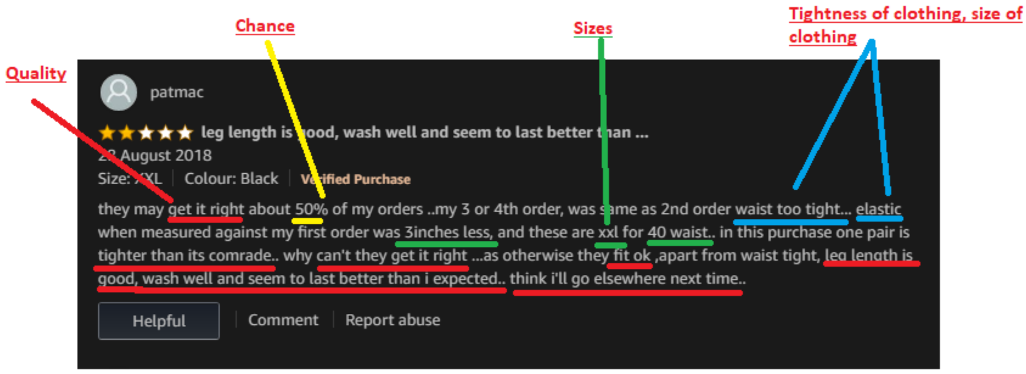


Figure 1.2: Example properties

on the metadata available from the site itself, but for the latter the way to obtain this information is less clear. Although we may infer that the rating has some indication of these properties, it does not describe the properties or the degree to which the review refers to them. This kind of information is valuable for making sense of the world of unstructured text, and has broad applications, e.g. The most immediate example is perhaps that they allow for a natural way to implement critique-based recommendation systems, where users can specify how their desired result should relate to a given set of suggestions [**?**]. For instance, [**?**] propose a movie recommendation system in which the user can specify that they want to see suggestions for movies that are "similar to this one, but scarier". If the property of being scary is adequately modelled as a direction in a semantic space of movies, such critiques can be addressed in a straightforward way. Similarly, in [**?**] a system was developed that can find "shoes like these but shinier", based on a semantic space representation that was derived from visual features. Semantic search systems can use such directions to interpret queries involving gradual and possibly ill-defined features, such as "*popular* holiday destinations in Europe" [**?**]. While features such as popularity are typically not encoded in traditional knowledge bases, they can often be represented as semantic space directions.

1.1.1 Directions

However, manually labelling these properties and the degrees to which entities (e.g. reviews, text-posts) have them is extremely time-consuming.

A potentially ideal system would be as follows: We collect large amounts of unstructured text data, separated into domains, and obtain the properties of each domain from this data, and rank entities on the degree to which they have these properties. In this way, properties would be understood on a scale built from the domain directly, so that each domain has its own meanings for words according to their own idiosyncrasies. As the process does not require any manual labelling the quality of these properties could be improved simply by obtaining more data. Further, as we are learning from unstructured data, not only would this allow us to understand the data in terms of what we know, but it would also introduce us to new ideas that we may not have previously understood. This kind of representation also has value in application to Machine Learning tasks. If we can separate the semantics of the space linearly into properties, we are able to learn simple linear classifiers that perform well.

Simple linear classifiers built from a representation composed of rankings on properties have an additional benefit of being more understandable.

1.2 Interpretability

Most successful approaches in recent times, like vector-spaces, word-vectors, and others, rely on the distributional model of semantics. This model relies on encoding unstructured text e.g. of a movie review, as a vector, where each dimension corresponds to how frequent each word is, we are able to calculate how similar the entities are, e.g. we know that if two movies have a similar distribution of words in their reviews, like frequent use of the word 'scary', or 'horror', then they would have a higher similarity value. These models, also known as 'semantic spaces' encode this similarity information spatially.

Semantic relationships can be obtained from semantic spaces.

applications/need for good interpretability:

- Safety
- Troubleshooting, bug fixing, model improvement
- Knowledge learning
- EU's "Right to explanation"
- Discrimination

properties of an interpretable classifier:

- Complexity: 'the magic number is seven plus or minus two' [10] also has many positive effects for its users, like lower response times [9, 7], better question answering and confidence for logical problem questions [7] and higher satisfaction [9].
- Transparancy:
- Explainability:
- Generalizability:

Properties, entities, the benefits and application of a representation formed of these

Basic introduction to directions, explanation of the utility and application of our approach

1.3 Thesis Overview / Contributions

In 3, we focus on further experimenting with one relationship that was formalized in [6]: a ranking of entities on properties. In particular, we use this method of building a representation of entities as a way to convert a vector space into an interpretable representation, for use in an interpretable classifier. The reason that we chose this representation to expand on is because by representing each entity e with a vector v that corresponds to a ranking r, the meaning of each dimension is distinct, and we are able to find labels composed of clusters of words for these dimensions. Here, we make the distinction between a property and a word, a property is a natural property of the space that exists in terms of a ranking of entities, and words are the labels we use to describe this property.

Chapter 2

Background

2.1 Text Representations

Need to write about the concept of salient features of a domain here.

2.1.1 Bag-of-words

We begin by processing an unstructured text corpus, composed of documents C_D. We then remove all punctuation, convert any accented characters to non-accented characters, and lowercase the documents to obtain word tokens for each document D_W. From here, we can assume that any $W \approx W$ will now $W = W$, if a word varied in format but not alphanumeric characters.

Then, we count the occurrences of each word

- Frequency
- Tf-idf
- PPMI

2.2 Text classification

2.2.1 Decision Trees

- Explanation of what decision trees are
- Explanation that they may not perform well on sparse information

2.2.2 Support Vector Machines

- Performance increase for support vector machines on sparse data, balancing, etc

2.2.3 Neural Networks

- Difference between SVM and Nnet

2.2.4 Semantic Spaces

Bag-Of-Words representations of text result in large sparse vectors for each document,

How do vector spaces represent semantics? Why do we use them to represent semantics?

Distributional representations of semantics, known as 'semantic spaces' are well-recognized for their ability to represent semantic information spatially. These representations have been widely adopted for Natural Language Processing (NLP) tasks thanks to their ability to represent complex information in a dense representation. In particular, entity-embeddings have been applied to represent items in recommender systems [**?**, **?**, **?**], to represent entities in semantic search engines [**?**, **?**], or to represent examples in classification tasks [**?**].

Vector spaces are a popular way to represent unstructured text data, and have been broadly applied to and transformed by supervised approaches. They vary in method, producing structure from Cosine Similarity, Matrix Factorization, Word-Vectors/Doc2Vec, etc. They also vary in how they linearly separate entities. However, their commonality is that they are able to represent semantic relationships spatially. See Section 2.2.4 This brings up an essential point: When using a semantic space, are we taking advantage of relationships that are discriminative or incorrect? The danger of relying on these spaces and the models that use them has greatly affected their adoption in critical application areas like medicine, and has raised legal concerns about their application in e.g. determining if someone is suitable for a loan.

See Section 2.2.4

- Word-vectors

2.2.5 Document Representations

LSA

Principal Component Analysis is a dimensionality reduction method that results in dimensions ordered by importance. Starting with a large data matrix, e.g. our TF-IDF values from before, we first find the covariance matrix for these values. Then, from this covariance matrix we obtain the eigenvalues. We can then linearly transform the old data in-terms of this covariance matrix to obtain a new space of size equal to an arbitrary value smaller than our matrix.

- PCA
- MDS

2.3 Interpretable Representations

a. NNSE b. compositional c. 2007 paper as wikipedia similarities d. Topic models e. Infogan, etc

[?] Sparse PCA (Why not compare lol)

Vector space models typically use a form of matrix factorization to obtain low-dimensional document representations. By far the most common approach is to use Singular Value Decomposition [?], although other approaches have been advocated as well. Instead of matrix factorization, another possible strategy is to use a neural network or least squares optimization approach. This is commonly used for generating word embeddings [?, ?], but can similarly be used to learn representations of (entities that are described using) text documents [?, ?, ?]. Compared to topic models, such approaches have the advantage that various forms of domain-specific structured knowledge can easily be taken into account. Some authors have also proposed hybrid models, which combine topic models and vector space models. For example, the Gaussian LDA model represents topics as multivariate Gaussian distributions over a word embedding [?]. Beyond document representation, topic models have also been used to improve word embedding models, by learning a different vector for each topic-word combination [?].

The most commonly used representations for text classification are bag-of-words representations, topic models, and vector space models. Bag-of-words representations are interpretable in principle, but because the considered vocabularies typically contain tens (or hundreds) of thousands of words, the resulting learned models are nonetheless difficult to inspect and understand. Topic models and vector space models are two alternative approaches for generating low-dimensional document representations.

2.3.1 Word Vectors

Chapter 3

Converting Vector Spaces into Interpretable Representations

3.1 Introduction

The ever more pervasive digital infrastructure that supports our lives has resulted in many opportunities to obtain data and models to make sense of that data. Semantic Spaces that encode semantic relationships between documents spatially have recently achieved strong results on tasks like X, Y, Z. These neural-network learned representations make use of a variety of new information like grammatical structure, word-context and even image data. Further, as domains become more entrenched in the digital world, the need for models in safety critical domains like medicine or legal domains like credit evaluation have increased the need for producing interpretable models, as well as interpretable representations. However, the dimensions of a semantic space do not correspond to human understandable features, and standard approaches to interpretable text representations do not match the performance of these methods. Ideally, we would obtain a representation that makes use of the rich semantic relationships from a high-performing semantic space, but also has dimensions corresponding to interpretable features. To this end, we aim to introduce in this chapter a methodology to linearly transform a semantic space using just its associated bag-of-words as input into an interpretable representation, and demonstrate the applicability of this interpretable representation to simple interpretable classifiers.

There are many types of semantic relationships in a semantic space. For our work, the representation is composed of rankings of documents on semantic directions in the space, in particular where those directions correspond to features. We show an example of the kind-of directions we use to obtain our representation in 3.1. Directions from domain-specific semantic spaces have been used previously in a variety of ways, For instance, [?] found that features of countries, such as their GDP, fertility rate or even level of CO_2 emissions, can be predicted from word embeddings using a linear regression model. Similarly, in [?] directions in word embeddings were found that correspond to adjectival scales (e.g. bad $<$ okay $<$ good $<$ excellent) while [?] found directions indicating lexical features such as the frequency of occurrence and polarity of words.

Derrac [6] introduced an unsupervised method to go from a semantic space and its associated bag-of-words to a representation where each dimension is a ranking of documents on a feature of the domain. For example, in the domain of movie reviews genres would be a feature, and the dimension would have a numeric value for each document corresponding to the degree it is a particular genre. The contribution of this Chapter is an analysis and experimentation on the quality of these features applied to document classification. The main insight from our work is that these interpretable features do not suffer a performance drop in a non-linear classifier compared to the original representation, and can outperform the original representation and a baseline interpretable representation in a linear classifier. In addition, we find that if a dimension ranks documents on a feature relevant to the task, it can be competitive with more complex models using a single decision tree node. We show an example of the representation from a domain of IMDB movie reviews in 3.2.

This chapter continues as follows: We begin by describing related work, then explain the method, making explicit the variations we have introduced for our experimental work. We follow this with the results of our experiments accompanied by qualitative

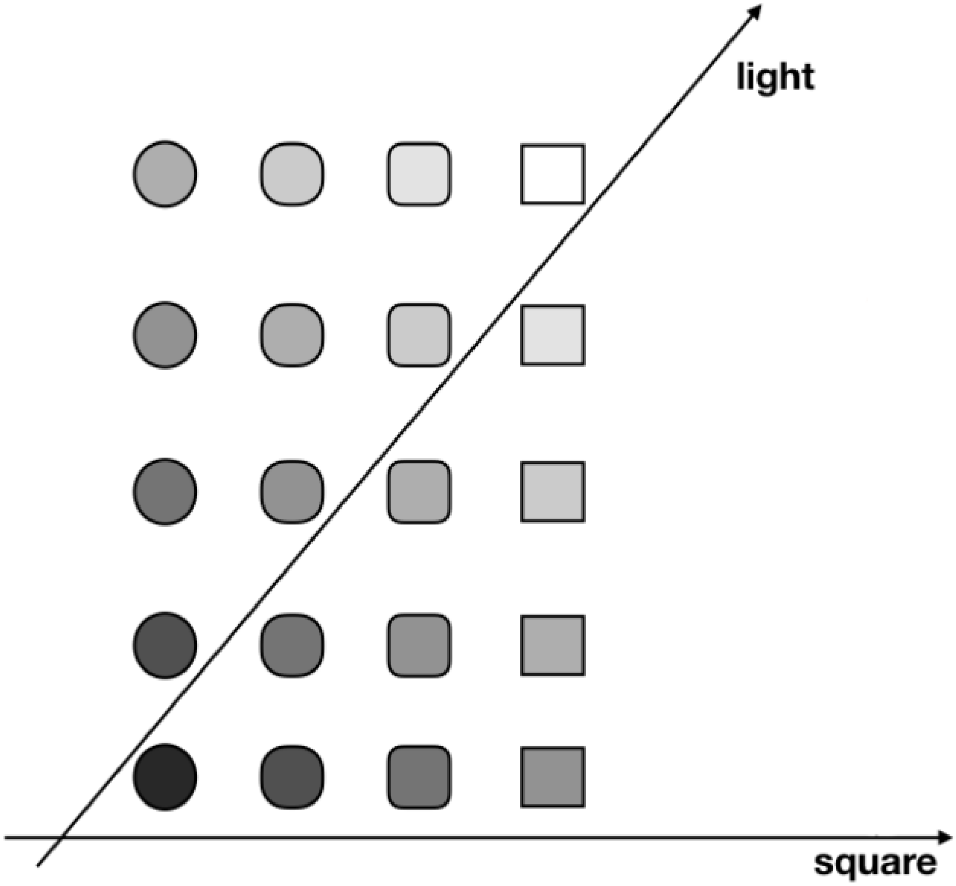


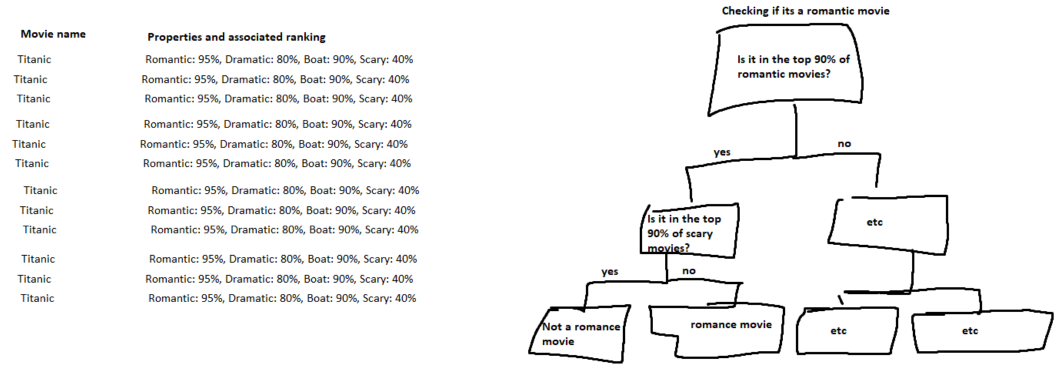
Figure 3.1: An example in a toy domain of shapes.

examples and explanations, and finish with a conclusion on the benefits and limitations of this approach.

3.2 Related Work

3.2.1 Semantic Relations & Their Applications

Our method uses the relationships inherent in a semantic space. Other work has formalized the relationships found in semantic spaces, for example in word-vectors, linear analogies (see Section **??**, were found where the vector between king and queen was and rep.png



Movie name	Properties and associated ranking
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%
Titanic	Romantic: 95%, Dramatic: 80%, Boat: 90%, Scary: 40%

Figure 3.2: Example movies and selected associated dimensions, chosen according to their relevance to the genre task..

parallel to the vector between prince and princess. These relationships have been expanded on, for example [11] found that "equivalent relations tended to correspond to parallel vector differences" [8], and [8], found that by decomposing representations into orthogonal semantic and syntactic subspaces they were able to produce substantial improvements on various tasks. Additionally, they have also been found to hold inherent gender bias [**?**] as word distances between gendered words (e.g. male, female, she, her) and occupational words e.g. (nurse, programmer) were correlated to the percentage of occupation that gender had for that role in different time periods. **Linear Classifiers** Decision trees, linear SVM's, logistic regression, decision tables, IF Then rules.

What are the available options for interpretable linear classification?

How have each of these methods been measured or validated in the literature in regards to interpretability? How about application to real world situations?

Non linear classifiers What non linear classifiers networks are interpretable? How have they done it? How have they measured it? How does it compare to a linear method?

*Neural networks*Approximating w/linear model, Interpretable nodes/weights

Other Stuff

3.2.2 Interpretable Representations

There are two ways in which topic models can be used for document classification. First, a supervised topic model can be used, in which the underlying graphical model is explicitly extended with a variable that represents the class label [4]. Second, the parameters of the multinomial distribution corresponding to a given document can be used as a feature vector for a standard classifier, such as a Support Vector Machine (SVM) or Decision Tree. LDA has been extended by many approaches, e.g. aiming to avoid the need to manually specify the number of topics [**?**], modelling correlations between topics [3], or by incorporating meta-data such as authors [**?**] or time stamps [**?**].

Broadly speaking, in the context of document classification, the main advantage of topic models is that their topics tend to be easily interpretable, while vector space models tend to be more flexible in the kind of meta-data that can be exploited. The approach we propose in this paper aims to combine the best of both worlds, by providing a way to derive interpretable representations from vector space models.

One of the more popular models for text representation that labels features in a similar way to our method are Topic Models.

3.3 Method

This section details the methodology to go from a Bag-Of-Words (BOW) 2.1.1 and Semantic Space 2.2.4, to rankings of documents on features of the domain, e.g. In a domain of IMDB movie reviews, where a document is composed of all of its reviews, a movie would be ranked on features like $Scary, Horror, Bloody$ and $Romantic, Love, Cute$, ideally with as many rankings as salient features of the domain.

3.3.1 Obtaining Directions and Rankings From Words

In this section we show how to obtain directions for words, and explain how to obtain document representations by ranking documents on these directions. For this step, we do not expect all words to be features of the domain. In the next sections, we aim to filter these words to obtain salient features.

Obtaining directions for each word For each word w, a Support Vector Machine (See Section 2.2.2) classifier is trained on the binary Bag-Of-Words representation of that word, where words are labelled as positive if they occurred more than once $w_f >= 1$ and negative otherwise. Although the separation of documents is binary, we can expect that the degree to which they are classified as the word varies. For example in a space constructed from frequency vectors, we can expect that the documents which contain the word more frequently would be further away from the hyper-plane in the positive direction. Following this, we can consider the vector v_w perpendicular to the hyperplane as the direction that models documents from least relevant at the distance furthest from the hyperplane on the negative side to most relevant for the word w at the distance furthest from the hyperplane at the positive side. We show an example of this in the toy domain in Figure 3.3.

Ranking documents on directions Once we have obtained a direction vector for each word v_w the next step is to quantify the degree to which each document has that word, by obtaining a value that corresponds to how far-up it is on the direction vector. These are our rankings of documents on words, if p_d is the representation of a document in the given vector space as a point then we can think of the dot product between the hyper-plane and the document vector $H_w \cdot p_d$ as the ranking $r_d w$ of the document d for the word w, and in particular, we take $r_d 1 < r_d 2$ to mean that d_2 has the property labelled with the word w to a greater extent than e_1. Below, we show some examples of features and documents ranked on them for different domains.

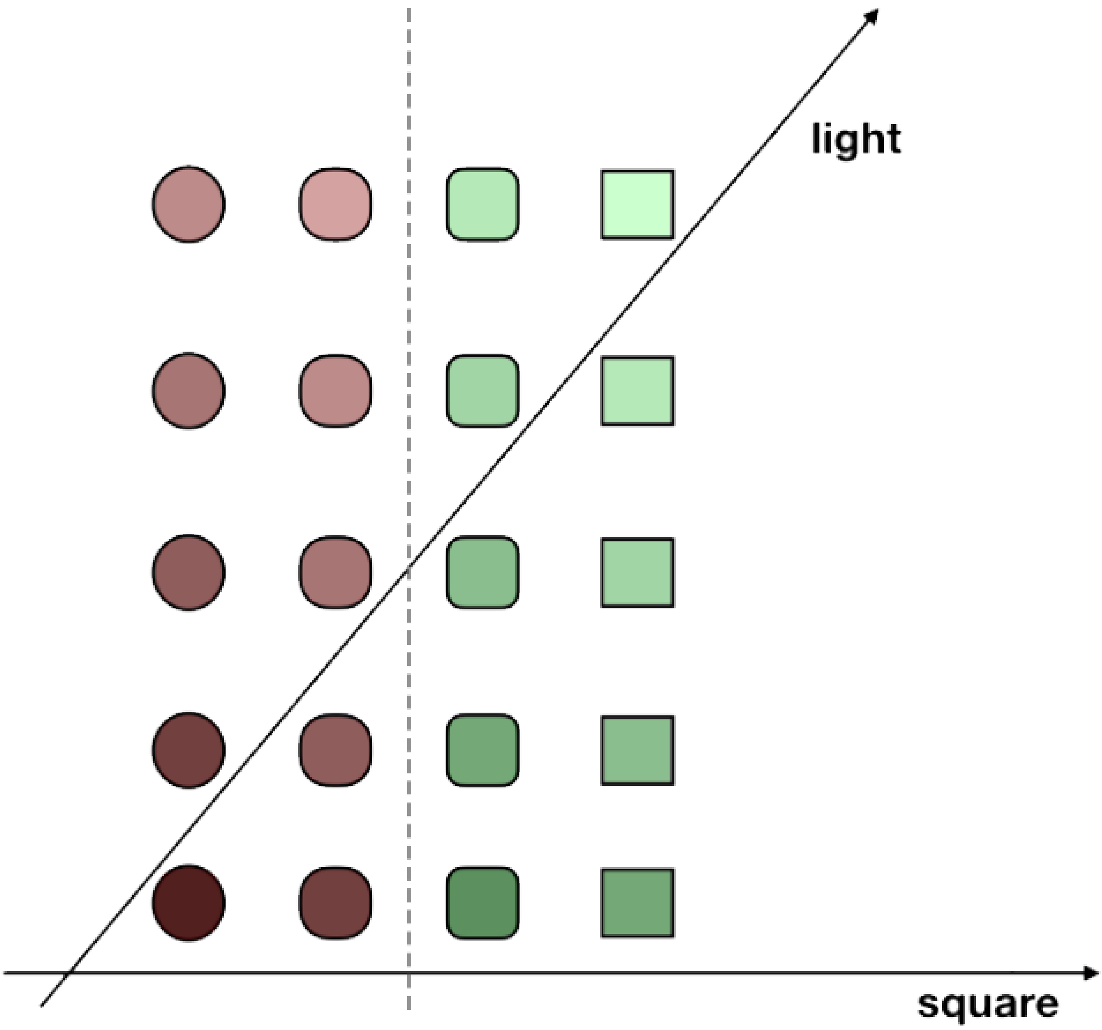


Figure 3.3: An example of a hyper-plane and its orthogonal direction in a toy domain of shapes. Green shapes are positive examples and red shapes are negative examples, but despite the problem being binary those closest to the hyper-plane are less defined than those further away, resulting in the orthogonal vector being a direction..

3.3.2 Filtering Words

With the rankings R_r, we could create a representation of each document d, composed of w_n dimensions, where each dimension is a ranking of the document d on that word $r_d w$. However, many of the words are not spatially important enough in the representation to result in a quality ranking - they are not salient features. In this section, we aim to filter the words that are not separable, we evaluate them using a scoring metric, and remove the words that are not sufficiently well scored. We use three different metrics:

Classification accuracy. Evaluating the quality in terms of the accuracy of the SVM classifier: if this classifier is sufficiently accurate, it must mean that whether word w relates to document d (i.e. whether it is used in the description of d) is important enough to affect the semantic space representation of d. In such a case, it seems reasonable to assume that w describes a salient property for the given domain.

Cohen's Kappa. One-kind of feature we find in these domains are binary labels of documents, for example a movie either is or isn't a movie with "Gore". We can expect that the more salient a binary feature, the more linearly separable it will be in the space. One problem with accuracy as a scoring function is that these classification problems are often very imbalanced. In particular, for very rare words, a high accuracy might not necessarily imply that the corresponding direction is accurate. For this reason, [6] proposed to use Cohen's Kappa score instead. In our experiments, however, we found that accuracy sometimes yields better results, so we retain Kappa as an alternative metric.

Normalized Discounted Cumulative Gain This is a standard metric in information retrieval which evaluates the quality of a ranking w.r.t. some given relevance scores [**?**]. In our case, the rankings r_d of the document d are those induced by the dot products $v_w \cdot d$ and the relevance scores are determined by the Pointwise Positive Mutual Information (PPMI) score *ppmi*(w, d), of the word w in the BoW representation of entity d where *ppmi*$(w, d) = \max\left(0, \log\left(\frac{p_{wd}}{p_{w*} \cdot p_{*d}}\right)\right)$, and

$$p_{wd} = \frac{n(w,d)}{\sum_{w'} \sum_{d'} n(w', d')}$$

where $n(w, d)$ is the number of occurrences of w in the BoW representation of object d, $p_{w*} = \sum_{e'} p_{wd'}$ and $p_{*d} = \sum_{w'} p_{w'd}$.

By scoring the words on these features, we can apply a simple cut-off (e.g. the top

2000 scored words) to obtain the most salient words. Ideally, this cut-off would be at the point where the words stop corresponding to salient features. However, it is difficult to determine this. In principle, we may expect that accuracy and Kappa are best suited for binary features, as they rely on a hard separation in the space between objects that have the word in their BoW representation and those that do not, while NDCG should be better suited for gradual features. In practice, however, we could not find such a clear pattern in the differences between the words chosen by these metrics despite often finding different words. In Table **??**, we show examples of the differences between the largest differences between the scoring methods.

Clustering Direction Vectors

If we consider two directions, "Blood" and "Gore", we can understand both of these to be approximating a similar feature of movies, as they both relate to how much blood a movie contains. Because of this, we can expect their directions to be very similar to each other. This is the first idea behind clustering these directions, if we average these directions together we can obtain a direction inbetween them that is a balance between documents that used the word 'Bloody' to describe the blood and the word 'Gore'. To expand on this, some entities would have the property of being bloody films, but did not necessarily use the term gore in their reviews, same as some entities having the property but using the term gore not bloody, we can understand that this new hyper plane and associated direction more accurately represents the property of a bloody film more than either of the terms individually. By extending this to a clustering method, we can find similar abstract features by ensuring that all similar directions are clustered together.

The word direction for "beautiful" can be nebulous to the interpreter, as it is not clear what it means for a movie to be ranked highly on 'beautiful'. Considering this, clustering provides another advantage, once we cluster the terms to find the property ("beautiful", "cinematography" "shots") we are given context for the word and more easily intuit the feature, in this case it is a feature about how well the movie was directed.

We approach clustering the directions with a variety of methods:

K-Means K-Means is a clustering algorithm that starts with determining the amount of clusters, K. To begin, K centroids c are randomly placed into the space. Then, the distance between each point p and centroid c (in our case, points are determined by rankings) is calculated. Each point p is then assigned to its closest centroid c. Then, the centroids are recomputed to be the mean of their assigned points. This process starting with the distance calculation is repeated until the points assigned to the centroids do not change.

Derrac's K-Means Variation This is the clustering method used in the previous work [**?**]. As input to the clustering algorithm, we consider the N best-scoring candidate feature directions v_w, where N is a hyperparameter. The main idea underlying their approach is to select the cluster centers such that (i) they are among the top-scoring candidate feature directions, and (ii) are as close to being orthogonal to each other as possible.

The output of this step is a set of clusters $C_1, ..., C_K$, where we will identify each cluster C_j with a set of words. We will furthermore write v_{C_j} to denote the centroid of the directions corresponding to the words in the cluster C_j, which can be computed as $v_{C_j} = \frac{1}{|C_j|} \sum_{w_l \in C_j} v_l$ provided that the vectors v_w are all normalized. These centroids $v_{C_1}, ..., v_{C_k}$ are the feature directions that are identified by our method.

We choose our first cluster centroid by taking the top-scoring direction for its associated metric. Then, we select centroids until we have reached the desired amount by taking the maximum of the summed absolute cosine similarity of all current centroids, in other words taking the most dissimilar direction to all of the current directions. Once we have selected the centroids, for each remaining direction we find the centroid it is most similar to, and the centroid is updated once the direction has been added.

3.3.3 Qualitative Results

The best-performing directions for each space type

Movies *(50 MDS NDCG)*	**Sentiment** *(100 D2V NDCG)*	**Newsgroups** *(50 D2V NDCG)*	**Place-types** *(50 PCA Kappa)*	**Reuters** *(200 MDS NDCG)*
horror (scares, scary)	glenda (glen, matthau)	karabag (iranian, turkiye)	blackcountry (listed, westmidlands)	franklin (fund, mthly)
hilarious (funniest, hilarity)	scarlett (gable, dalton)	leftover (flaming, vancouver)	ears (stare, adorable)	quarterly (shearson, basis)
bollywood (hindi, india)	giallo (argento, fulci)	wk (5173552178, 18084tmibmclmsuedu)	spagna (espanha, colores)	feb (28, splits)
laughs (funnier, funniest)	bourne (damon, cusack)	1069 (mlud, wibbled)	oldfashioned (winery, antiques)	22 (booked, hong)
jokes (gags, laughs)	piper (omen, knightley)	providence (norris, ahl)	gardening (greenhouse, petals)	april (monthly, average)
comedies (comedic, laughs)	casper (dolph, damme)	celestial (interplanetary, bible)	pagoda (hindu, carved)	sets (principally, precious)
hindi (bollywood, india)	norris (chuck, rangers)	mlud (wibbled, 1069)	artificial (saturation, cs4)	16 (creditor, trillion)
war (military, army)	holmes (sherlock, rathbone)	endif (olwm, ciphertext)	inner (curved, rooftops)	1st (qtr, pennsylvania)
western (outlaw, unforgiven)	rourke (mickey, walken)	gd3004 (35894, intergraph)	celebrate (festive, celebrity)	26 (approve, inadequate)
romantic (romance, chemistry)	ustinov (warden, cassavetes)	rtfmmitedu (newsanswers, ieee)	vietnamese (ethnic, hindu)	23 (offsetting, weekly)
songs (song, tunes)	scooby (doo, garfield)	eng (padres, makefile)	cn (elevated, amtrak)	prior (recapitalization, payment)
sci (science, outer)	doo (scooby, garfield)	pizza (bait, wiretap)	mannequin (bags, jewelry)	avg (shrs, shr)
funniest (hilarious, funnier)	heston (charlton, palance)	porsche (nanao, mercedes)	falcon (r, 22)	june (july, venice)
noir (noirs, bogart)	homer (pacino, macy)	gebcadredslpittedu (n3jxp, skepticism)	jewish (monuments, cobblestone)	march (31, day)
documentary (documentaries, footage)	welles (orson, kane)	scsi2 (scsi, cooling)	canon60d (kitlens, 600d)	regular (diesel, petrol)
animation (animated, animators)	frost (snowman, damme)	playback (quicktime, xmotif)	reflective (curved, cropped)	4th (qtr, fourth)
adults (adult, children)	streisand (bridget, salman)	35894 (gd3004, medin)	mason (edward, will)	27 (chemlawn, theyre)
creepy (spooky, scary)	davies (rhys, marion)	diesel (volvo, shotguns)	aerialview (manmade, largest)	14 (borrowing, borrowings)
gay (gays, homosexuality)	cinderella (fairy, stepmother)	evolutionary (shifting, hulk)	shelf (rack, boxes)	11 (chapter, ranged)
workout (intermediate, instruction)	boll (uwe, belushi)	techniciandr (obp, 144k)	monroe (raleigh, jefferson)	may (probably, however)
thriller (thrillers, suspense)	rochester (eyre, dalton)	8177 (obp, 144k)	litter (fujichrome, e6)	38 (33, strong)
funnier (laughs, funniest)	edie (soprano, vertigo)	shaw (medicine, ottoman)	streetlights (streetlamp, headlights)	m1 (m2, m3)
suspense (suspenseful, thrillers)	scarecrow (zombies, reese)	scorer (gilmour, lindros)	carlzeiss (f2, voigtlander)	dlr (writedown, debt)
arts (hong, chan)	kramer (streep, meryl)	xwd (xloadimage, openwindows)	manmade (aerialview, below)	five (years, jones)
christianity (religious, religion)	marty (amitabh, goldie)	ee (275, xloadimage)	demolished (neglected, rundown)	bushels (soybeans, ccc)
musical (singing, sing)	columbo (falk, garfield)	com2 (com1, v32bis)	wald (berge, wildflower)	revs (net, 3for2)
gore (gory, blood)	kidman (nicole, jude)	examiner (corpses, brass)	arquitetura (exposition, cidade)	29 (175, include)
animated (animation, cartoon)	juliet (romeo, troma)	migraine (ama, placebo)	greyscale (highcontrast, monochromatic)	acquisition (make, usairs)
gags (jokes, slapstick)	garland (judy, lily)	parliament (parliamentary, armored)	alameda (monday, marin)	payable (div, close)
sexual (sexually, sexuality)	hawn (goldie, matthau)	manhattan (bobbeviceicotekcom, beauchaine)	button (monday, only)	13 (dlrsbbl, groups)

Table 3.1: Table

Start chucking in real world examples here.

Do certain spaces perform better across multiple domains? What are the differences between spaces in terms of the features they represent? How does that relate to their domain performance? What are the differences between score types? How does that relate to their domain performance? How does performance on pure representation dimensions relate to performance on single dirs/clusters? What kind of similarities are there between domains? What kind of differences? What similarities are there between score types? What are the differences in directions between domains? Do we find different kinds of properties? How does it relate to vocabulary size or document amount? How do the directions in particular between a doc2vec space, trained on context, differ from one trained on PPMI?

What are the domain(s?) that best conveys the differences and similarities between different space types? Movies, unless... Do different domains have more differences between space types? Theories: 1. Large differences in scoring for a particular domain will be indicative

1. Find the domain which has the largest score differences between two reasonably scoring space types (so not AWV..) 2. Compare and contrast the directions found in those space types.

What are the domain(s?) and space type(s?) that best conveys the differences and similarities between different score types?

1. Find the domain with the largest differences between score types 2. Compare and contrast directions in those score types for that domain

What are the domains that best convey the similarities and differences between different domains?

1. Find domains that act differently (perhaps one domain where a space-type that is not usually scoring high is scoring high, big differences in F1) 2. Get interesting directions from those domains

What are the differences and similarities between the clustering algorithms?

1. Find domain with the largest difference between clustering options

IF directions are non informative, then use clustering to label them

3.3.4 How Domain Directions Differ

Comparing IMDB Movie Reviews to Concatenated Movie Reviews from a variety of sources

3.3.5 Comparing Space Types

We begin by selecting the space that performed well on the genres task for the movies, with the understanding that genres as a key natural classification task will likely make use of good directions that correspond to domain knowledge. After selecting this space, we choose similarly sized spaces from the other space-types, in this case we selected the 50 dimensional MDS space as it performed the best and from there, we selected the 50 dimensional PCA space and AWV space. We approach these qualitative experiments with the following idea: spaces that perform better on natural domain tasks using decision trees contain unique natural directions that other spaces do not have.

The commonalities between spaces are much more prevalent than the differences, with natural concepts of the domain being represented in all of the different space types. However, different spaces do perform better than others on natural domain tasks. In this section, we investigate why this occurs and the differences between spaces built using a standard frequency-based approach, word-vectors and doc2vec, which uses a combination of contextual information and word vectors.

Comparing reps constructed from PPMI scores (MDS and PCA)

We show terms unique to each domain type above. The terms are contextualized by finding the two closest term directions to the term. Here, we show the terms that are not duplicated in meaning for the other spaces. We can understand that the term 'gay (gays, homosexuality) has a similar meaning to the term 'homosexuality (gays, gay)' despite the term 'homosexuality' being high scored for one space, and the term 'gay' being highly scored for the other space. Almost all of the unique terms were of this variety. To demonstrate instead the unique meanings found in the individual spaces, we filter the results as follows: from the term and its associated descriptive terms found by getting the most similar terms, if those terms or any of the more similar terms occur in any of the terms or more similar terms in the space's top 1000 terms that we are comparing to, do not include those terms.

We find that the 50-dimensional MDS space, which performs 0.04 F1 score higher on the genres task, finds many interesting and unique terms, which can potentially enable more nuanced decisions in the decision trees for classification. On the other hand, in the PCA space, we find terms that relate to metadata, and spanish language words. We argue that this means the MDS is better at finding unique interesting meanings than PCA, in the case of using a frequency-based BOW to create the representation.

Comparing MDS and AWV

We perform the same process as above but comparing MDS and AWV, and find that the PPMI-based representations have their own metadata in the representation that is elevated. We can assume that the AWV space does not contain these metadata as there are simply not word-vectors available for these terms. Although, we also find unique terms that are likely to help performance on natural domain tasks for the MDS space, e.g. goodfellas, disgusting, swashbuckling. Interestingly, for the AWV space we find similar spanish-language terms, but also find some new concepts, in particular the 'republican' and 'yoga' directions.

Comparing PPMI representations to doc2vec

The previous two experiments were conducted with the IMDB movies dataset, but the doc2vec space is not available for that dataset, as the original text corpus was not made available by the authors. Because of this, we choose to compare the PCA representation for the sentiment space and the Doc2Vec representation for the sentiment space, as these are still IMDB movie reviews, but instead with reviews as documents rather than compilations of reviews as documents, we can expect different directions to be important but still use our knowledge of movies and reviews to inform us, which would be difficult for the newsgroups or reuters representations which are mostly composed of archaic syntax. We take the 100 dimensional D2V space and the 100 dimensional PCA space.

The assumption going into this analysis was that, in the same way that the AWV space does not contain metadata that is not present in the word-vectors, as will the Doc2Vec space not be informed especially by that metadata. Further, as in this case the D2V space outperforms the PCA space on the sentiment task, we also expect that by using word-context to produce the representation, we are able to find better sentiment information directions, for example by better understanding sarcasm. Additionally, the benefit of word-vectors is likely to play into this space in a positive way, informing the representation based on global context.

When looking at these directions, we can split the terms into two types. The names of things, and conceptual properties about movies. For the doc2vec space, the names largely dominate the unique terms list, which we can understand to mean that these terms used contextually and in the broader global context from the word-vectors is informative enough in the doc2vec space to be usefully represented. For the PCA space, there are still these unique names found, but there are less overall terms and it's a bit more balanced between concepts and names.

As this was not particularly conclusive, we additionally compare the MDS representation for the newsgroups and the Doc2Vec representation. We chose the newsgroups in this case because the matters discussed on those newsgroups can be summed up into meaningful domain knowledge, and are not as difficult to interpret for a layman to the field as the reuters newswires, which are dominated by syntax and numbers.

3.3.6 Datasets

We use five

3.3.7 Producing Semantic Spaces

We use unsupervised representation learning methods, with the intention to obtain a representation that represents all salient features of the domain and can adapt to a variety of tasks.

For the semantic space, we compute the Positive Pointwise Mutual Information (See **??**) scores for the Bag-Of-Words, and use that as input to a variety of different off-the-shelf dimensionality reduction algorithms. We explain these in further detail in Section **??**.

Dimension of the Space

Space-type

Scoring method

Clustering method

The effect of dimensions

The effect of space-type

The effect of scoring method

3.3.8 Quantitative Results

From a domain, e.g. movie reviews, where each document is a collection of reviews for a movie, we preprocess the text such that it is converted to lower-case, and non-alphanumeric characters are removed. From here, we remove standard English stop words using the NLTK library [**?**]. We show an example of a review's original and converted formats in Figure **??**. From this preprocessed corpus, we obtain a Bag-Of-Words where we count the frequency of each term BOW_wf, see 2.1.1.

The difference between single directions and clusters is best highlighted when comparing their use in simple interpretable classifiers. In figure **??** we demonstrate this.

1. Negative directions (e.g. church for horror) 2. Non-contextualized, non-direct ways of classifying, versus clustering which finds salient properties which almost directly correspond to these natural tasks.

Datasets

Vocabulary size/entity size/origin of data/classes

Semantic Spaces

In this section, we explain how we obtained four different Semantic Spaces.

As the newsgroups contained empty documents after removing all words that do not occur in at least 2 documents, we have removed these empty documents, leaving us

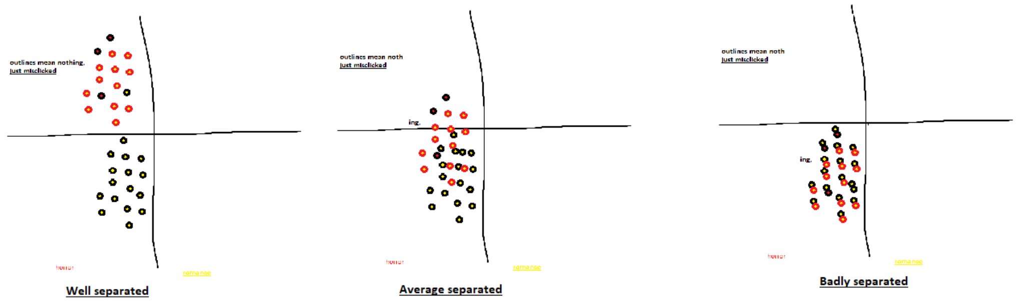


Figure 3.4: A conceptual space of movies, where regions correspond to properties and entities are points..

with 18302 overall documents. Following this, instead of using the train split as determined by previous literature, we did a simple 2/3 train/test split the same as our IMDB dataset.

This section focuses on using linear classifiers to determine how well our method represents domain knowledge compared to standard baselines. We can understand that an accurate representation of domain knowledge will be one that ensures semantically distinct entities are separated, and semantically similar entities are close together. Put another way, if the space is representing domain knowledge well we can expect that the space should be linearly separable for key semantics of the domain. For example, a good vector space in the domain of movies constructed from IMDB movie reviews should contain a natural separation of entities into genres, where Horror movies are spatially distant from Romance movies, and movies that are Romantic Horrors would be somewhere inbetween. We can see an example in Figure 3.4. For a Bag-Of-Words, we can expect similar entities to have similarly scoring terms **??**.

When selecting the parameters to use for the doc2vec space when obtaining directions, we choose the one that scored the highest for its class on a Linear SVM, rather than tuning the entire process around the doc2vecs vectors. We are not able to obtain an MDS space for sentiment or doc2vec spaces for placetypes/movies.

We obtain results with just these spaces as input, and additionally results for a bag-of-

	Top PPMI scoring terms
Example Horror Entity	Term term term term term term term term term term term term term
Similar Horror Entity	Term term term term term term term term term term term term term
Somewhere Inbetween Entity	Term term term term term term term term term term term term term
Romance Movie	Term term term term term term term term term term term term term
Similar Romance movie	Term term term term term term term term term term term term term

Table 3.2: Two of the following entities: Those classified as horror, those classified as horror and romance, and those classified as romance with their associated highest value PPMI terms. We show the highest positive instances here as the representation is sparse, even though we can also expect the terms that are low scoring to be similar too..

words PPMI representation. These results act as our baselines for quantitative results, in addition to a Topic Model. We find results using a Linear SVM, and a Decision Tree with an unlimited depth, a depth of 3, a depth of 2, and a depth of 1. Each SVM is tuned using a grid search for the optimal C value, and whether or not to balance the classes. For all trees we attempt to find the best value between [None, 'auto', 'log2'], and additionally try differnet criterion, either the gini score or the information entropy score. In the same way as the SVM's, we include whether or not to balance the classes in the grid search.

Word Directions

The binary BOW representation for each word that has not been removed by the frequency cut-off is used as a target for a linear SVM, with a Semantic Space as features. We use the scikit-learn libraries LinearSVC implementation with a default C value (1.0). We balance the classes, as many of the binary BOW representations are sparse, and use the primal formulation.

We obtain results for the rankings induced from these word directions on Decision

Tree's limited to a depth of 3 in-order to select the best parameters when using directions for each class. The parameters that we want to determine are the type of Semantic Space, the size of the space, the frequency threshold and the score threshold. To do so, for each space-type of each size, we use a grid search to find the best frequency and score cut-offs for that sized space-type. Then, we select from these space-types and sizes the best performing one. We can understand there to be a balance between finding words which are useful for creating salient features in our clustering step without including too many words which do not. As our clustering methods are unsupervised, it is important that we try and limit the amount of junk being entered into them, despite the classifiers that use these directions typically being able to filter out those directions which are not suitable to the class. Additionally, as the vocabulary size varies from dataset to dataset, the threshold will naturally be different for each one.

These results allow us to choose for each class, the best Semantic Space and Scoring-type for that class. For all trees we use grid search to find the best values for the criterion, either the gini score or the information entropy score, the maximum amount of features between [None, 'auto', 'log2'], and additionally, we include whether or not to balance the classes in the grid search.

Next, we test single directions, attempting to find a good amount of directions to cluster and not including words which may hamper the unsupervised classification, as well as the best space-type for each domain. We found that generally, X was the best space and as expected classifiers performed better with more data, so we use 20000 as our frequency cutoff and 2000 as our score cutoff. These single directions typically overfit.

Clustered Directions

We continue with the optimal space and score-type chosen by our single direction experiments, and use the same frequency and score thresholds as before. We then experiment with two different clustering algorithms: Derrac and K-Means. As these algorithms select centroids from the top-scoring directions or randomly, we can expect that some clusters may not be salient features of the space. This is because top-scoring directions, e.g. for accuracy could simply infrequent terms that do not have much meaning, and these infrequehnt terms could also be randomly selected. We could use grid-search on the frequency and score cutoffs when obtaining these results in order to avoid terms that may disrupt existing clusters or form cluster centers that are not salient features of the space, but we chose a more standardized process that would rely on the parameters of the clustering algorithms and the ability of the classifiers to filter out clusters that are not informative, so as to not make a time-costly grid search a necessary part of the process.

With that in mind, we use three clustering algorithms.

Mini batch K-means, implemented by scikit-learn [1], introduced by [**?**] and kmeans++ to initialize [**?**]

Quantitative Results

In Figure 3.5, we demonstrate how depth could affect a Decision Tree that uses salient feature-directions.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html

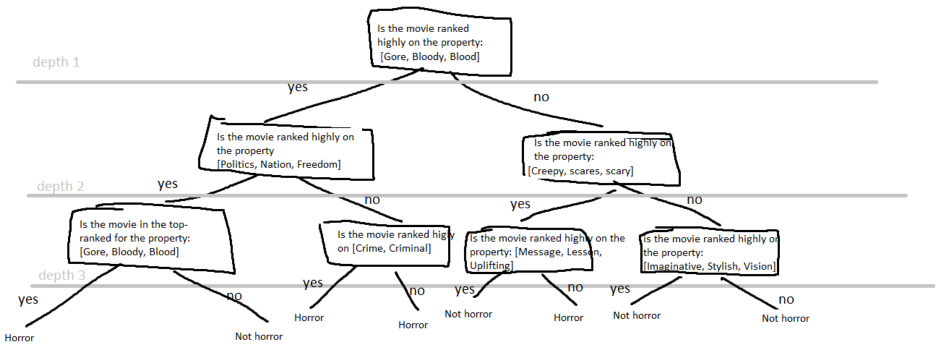


Figure 3.5: This figure shows an example tree from one of our classifiers. Here, we can see that the model increases in complexity as it increases in depth. In this case, we end-up getting better F-score with just a depth-one tree, as the tree begins to overfit at depth three. .

Movies	Genres			Keywords			Ratings		
	D1	D2	D3	D1	D2	D3	D1	D2	D3
Space	0.301	0.358	0.354	0.185	0.198	0.201	0.463	0.475	0.486
Single directions	0.436	0.463	0.492	0.230	N/A	N/A	0.466	N/A	N/A
Clusters	N/A	N/A	0.518	N/A	N/A	N/A	N/A	N/A	N/A
	Newsgroups			Sentiment			Reuters		
Rep	0.251	0.366	0.356	0.705	0.770	0.773	0.328	0.413	0.501
Single dir	0.418	0.490	0.537	0.784	0.814	0.821	0.678	0.706	0.720
Cluster	0.394	0.433	0.513	0.735	0.844	0.813	0.456	0.569	0.583
	Foursquare			OpenCYC			Geonames		
Placetypes	D1	D2	D3	D1	D2	D3	D1	D2	D3
Rep	0.438	0.478	0.454	0.383	0.397	0.396	0.349	0.340	0.367
Single dir	0.541	0.498	0.531	0.404	0.428	0.390	0.444	0.533	0.473
Cluster	0.462	0.507	0.496	0.413	0.420	0.429	0.444	0.458	0.470

Table 3.3: Best-scoring results for each type.

3.3.9 Interpretability Results

Chapter 4

Fine-tuning Vector Spaces to Improve Their Directions

"Commonly, these representations are made in a single vector space with similarity being the main structure of interest. However, recent work by Mikolov et al. (2013b) on a word-analogy task suggests that such spaces may have further use- ful internal regularities. They found that seman- tic differences, such as between big and small, and also syntactic differences, as between big and bigger, were encoded consistently across their space. In particular, they solved the word-analogy problems by exploiting the fact that equivalent re- lations tended to correspond to parallel vector- differences. [8]

[8] "Explicitly designing such structure into a neural network model results in rep- resentations that decompose into orthog- onal semantic and syntactic subspaces. We demonstrate that using word-order and morphological structure within En- glish Wiki- pedia text to enable this de- composition can produce substantial im- provements on semantic-similarity, pos- induction and word-analogy tasks."

This means that despite state-of-the-art results in Natural Language Processing tasks like Language Modelling, Machine Translation, Text Classification, Natural Language Inference, Abstractive Summarization, and Dependency Parsing being dominated by neural networks that learn and improve these kind-of representations, it is not clear what information has been represented.

4.1 Experiments

We find that non-linearity is useful.

Chapter 5

Investigating Neural Networks In Terms Of Directions

5.1 Appendix

5.1.1 Chapter 3 Space Types

	Genres			Keywords			Ratings		
Movies	D1	D2	D3	D1	D2	D3	D1	D2	D3
Space	50 PCA	50 MDS	100 MDS	200 PCA	200 MDS	200 MDS	50 PCA	200 PCA	50 PCA
Single directions	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Newsgroups			Sentiment			Reuters		
Rep	200 PCA	200 PCA	100 PCA	PCA 100	PCA 50	PCA 50	200 PCA	200 PCA	100 PCA
Single dir	200 MDS	100 D2V	50 D2V	D2V 100	PCA 50	D2V 100	N/A	N/A	N/A
	Foursquare			OpenCYC			Geonames		
Placetypes	D1	D2	D3	D1	D2	D3	D1	D2	D3
Rep	MDS 100	AWV 50	MDS 200	AWV 50	MDS 200	AWV 50	MDS 50	MDS 50	AWV 200
Single dir	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 5.1: Space-types, clusters have the same as single directions.

GNU Free Documentation License

0. Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. Applicability and Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. Copying in Quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. Modifications

you may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © `YEAR YOUR NAME`. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free

> Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

> with the Invariant Sections being `LIST THEIR TITLES`, with the Front-Cover Texts being `LIST`, and with the Back-Cover Texts being `LIST`.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2012.

[2] David M Blei, Blei@cs Berkeley Edu, Andrew Y Ng, Ang@cs Stanford Edu, Michael I Jordan, and Jordan@cs Berkeley Edu. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] David M. Blei and John D. Lafferty. Correlated Topic Models. *Advances in Neural Information Processing Systems 18*, pages 147–154, 2006.

[4] David M. Blei and Jon D. McAuliffe. Supervised Topic Models. pages 1–8, 2010.

[5] Jonathan Chang, Sean Gerrish, Chong Wang, and David M Blei. Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems 22*, pages 288—-296, 2009.

[6] Joaqu??n Derrac and Steven Schockaert. Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning. *Artificial Intelligence*, 228:66–94, 2015.

[7] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

[8] Jeff Mitchell and Mark Steedman. Orthogonality of Syntax and Semantics within Distributional Spaces. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1301–1310, 2015.

[9] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. pages 1–21, 2018.

[10] T.L. Saaty and M.S. Ozdemir. Why the magic number seven plus or minus two. *Mathematical and Computer Modelling*, 38(3):233–244, 2003.

[11] Geoffrey Zweig Tomas Mikolov , Wen-tau Yih. Linguistic Regularities in Continuous Space Word Representations. *Hlt-Naacl*, (June):746–751, 2013.