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 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

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

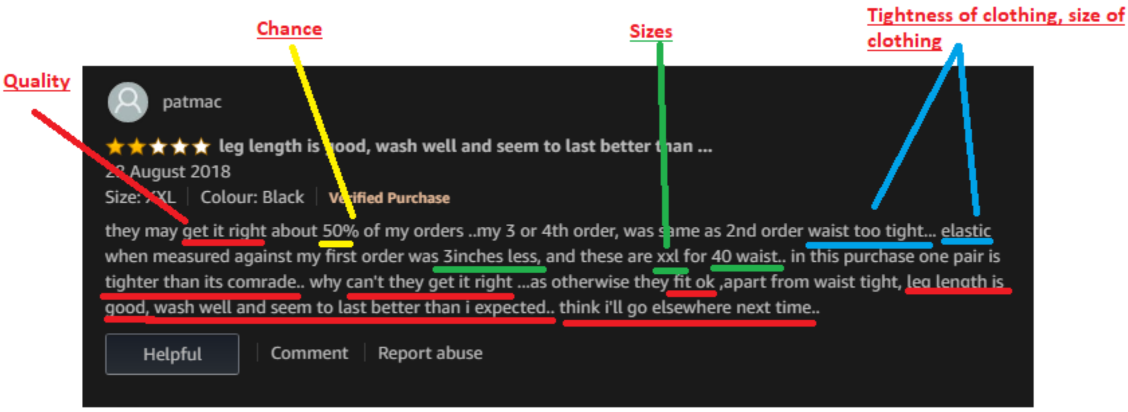

Figure 1.2: Example properties

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 [56]. For instance, [57] 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 [27] 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" [23]. 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' [46] also has many positive effects for its users, like lower response times [41, 20], better question answering and confidence for logical problem questions [20] and higher satisfaction [41].
- 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 [11]: 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 lower-case 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
- Max features
- Criterion
- CART decision trees versus others

2.2.2 Support Vector Machines

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

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 [54, 32, 52], to represent entities in semantic search engines [23, 53], or to represent examples in classification tasks [9].

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.

Dimensionality Reduction Methods

- PCA
- MDS

2.3 Interpretable Representations

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

[61] 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 [8], 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 [38, 43], but can similarly be used to learn representations of (entities that are described using) text documents [5, 52, 23]. 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 [7]. Beyond document representation, topic models have also been used to improve word embedding models, by learning a different vector for each topic-word combination [34].

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

This chapter introduces a methodology to go from a Vector Space Model (VSM) of text-documents and associated Bag-Of-Words (BOW) to an interpretable document representation. These interpretable representation are demonstrated to be effective in simple linear classifiers on document classification tasks. In this work, a representation is defined as ’interpretable’ if each feature is labelled and corresponds to a salient feature of the domain. In the case of the document representations used in this work, salient features correspond to rankings of documents on domain knowledge. Examples of features that rank documents from the method are shown in Table 3.1.

The method is entirely unsupervised and ’disentangles’ an existing vector space model into salient features. The idea of disentanglement is present in representation learning [2] e.g. when given a raw video file of a person jumping, ideally a representation would spatially separate the notions of ’jumping’, from the ’person’, and the ’background’. In this work disentanglement is used to refer to separating an existing vector space into salient features, such that each dimension of the space is a labelled feature.

Domain-specific semantic spaces like the ones in this work are used, for instance, to represent items in recommender systems [54, 32, 52], to represent entities in semantic search engines [23, 53], or to represent examples in classification tasks [9]. Methods like Multi-Dimensional Scaling and Principal Component Analysis that produce these semantic spaces are in wide-

IMDB Movie Reviews	Flickr-Placetypes	20-Newsgroups
courtroom legal trial court	broadway news money hollywood	switzerland austria sweden swiss
disturbing disgusting gross	fir bark activism avian	ham amp reactor watts
tear cried tissues tears	palace statues ornate decoration	karabag armenian karabakh azerbaijan
war soldiers vietnam combat	drummer produce musicians performers	4800 parity 9600 bps
message social society issues	ubahn railways electrical bahn	xfree86 linux
events accuracy accurate facts	winery pots manor winecountry	umpires umpire 3b viola
santa christmas season holiday	steeple religion monastery cathedral	atm hq ink paradox
martial arts kung	blanket whiskers fur adorable	lpt1 irq chipset mfm
bizarre weird awkward	desolate eerie mental loneliness	manhattan beauchaine bronx queens
drug drugs dealers dealer	carro shelby 1965 automobiles	photoshop adobe
inspirational inspiring fiction narrative	relax dunes tranquil relaxing	reboost fusion astronomers galactic

Table 3.1: Example features of our interpretable representation from three different domains. Each row is a label for a feature from our representation for that domain.

spread use for document representation and data analysis, and are typically built from word frequency statistics (See 2.2.5). However, a wide variety of training methods have been used to obtain semantic spaces. Distributional word-vectors that rely on learning via word-context have had great success as a component of neural learning systems achieving state-of-the-art results on key natural language processing tasks like Language Modelling [16], Constituency Parsing [13], and Part-Of-Speech Tagging [13], and have also been applied for document representations [30, 29]. Ideally, we would like to retain the benefits of these learning methods while also making them interpretable. The methodology in this work is a post-processing step that can be applied to representations regardless of how they have been learned, by leveraging the spatial relationships in the representation.

The spatial structures of semantic spaces have been used in a variety of ways. In [6] it was shown that it is possible to perform vector operations on Paragraph Vectors, e.g. subtracting word-vectors from paragraph vectors, like in the case of a corpus of arxiv papers, a paper titled "Spectral Clustering", could have the word-vector for "Spectal" subtracted from it to get papers about general clustering. In the case of distributional representations of words [51] found that "equivalent relations tended to correspond to parallel vector differences" [39], found that by decomposing representations into orthogonal semantic and syntactic subspaces they were able to produce substantial improvements on various tasks. In [25] directional vectors in word embeddings were found that correspond to adjectival scales (e.g. bad $<$ okay $<$ good $<$ excellent) while [45] found directions indicating lexical features such as the frequency of occurrence and polarity of words.

The spatial structures we leverage in this work are found in document representations. In particular, directional vectors that describe a particular feature of a domain. A toy example is shown in Figure 3.1. These directions have been applied in a variety of domains. For instance, [17] 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. Derrac [11] found directions corresponding to properties such as 'Scary', 'Romantic' or 'Hilarious' in a semantic space of movies, for example a direction which goes from a movie that is the least 'Scary' to the most 'Scary'. This chapter builds on their work, with the main contributions of the chapter being application of this method to producing an interpretable representation, deeper and more extensive experimentation, qualitative analysis and application to interpretable classifiers.

Such feature directions are useful in a wide variety of applications. 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 [55]. For instance, [58] 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 [26] 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" [22]. While features such as popularity are typically not encoded in traditional knowledge bases, they can often be represented as semantic space directions. As another application, feature directions can also be used in interpretable classifiers. For example, [11] learned rule based classifiers from rankings induced by the feature directions.

The simple linear classifiers that are used to evaluate the method's feature directions are low-depth Decision Trees. In Figure 3.2 an example is shown of a shallow Decision Tree using the method's interpretable representation. Shallow Decision Trees were chosen because they are effective at evaluating the disentanglement of the representations features. If the features are disentangled, then a low-depth Decision Tree will suffice to classify natural domain tasks. Shallow trees also evaluate the semantic generalizability of the features, as if they are able to

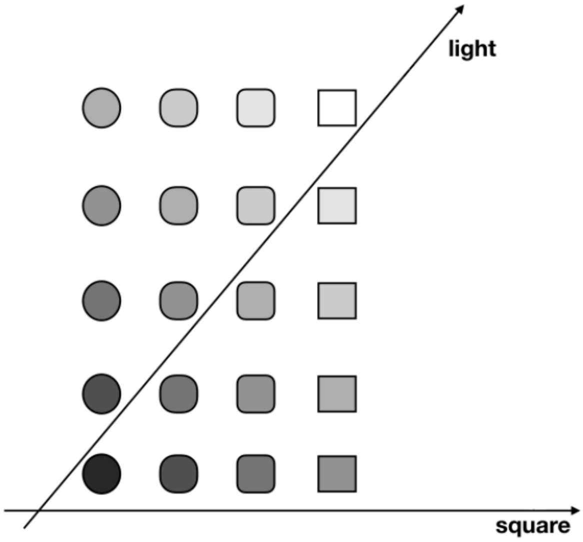

Figure 3.1: An example in a toy domain of shapes.

classify complex classes using only a single feature then that feature must be semantically coherent and generalizable. In terms of interpretability, shallow trees have many positive effects for users, like lower response times [41, 20], better question answering and confidence for logical problem questions [20] and higher satisfaction [41]. Although in this work the superiority of low-depth Decision Trees in real-world interpretability applications is not in the scope of the evaluation, as the interpretable representations could be applied to a variety of classifiers.

The quantitative results for the method results show that the method can successfully disentangle a variety of representations even with trees as limited as depth one, and these shallow trees outperform the original representation greatly when compared to deeper trees on the uninterpretable original features. Additionally, the results in most cases are also competitive with Latent Dirchlet Allocation, a baseline interpretable topic model. The method is shown to be an effective way to obtain a disentangled representation that can effectively produce simple interpretable classifiers. The method is verified to work on five different representation types for five different domains, using natural domain tasks for those domains.

The interpretable representation that is obtained by this method is composed of in terms of salient features, where each of these features is described using a cluster of natural language terms. This is somewhat similar to Latent Dirichlet Allocation (LDA), which learns a representation of text documents as multinomial distributions over latent topics, where each of these topics corresponds to a multinomial distribution over words [4]. Topics tend to correspond to salient

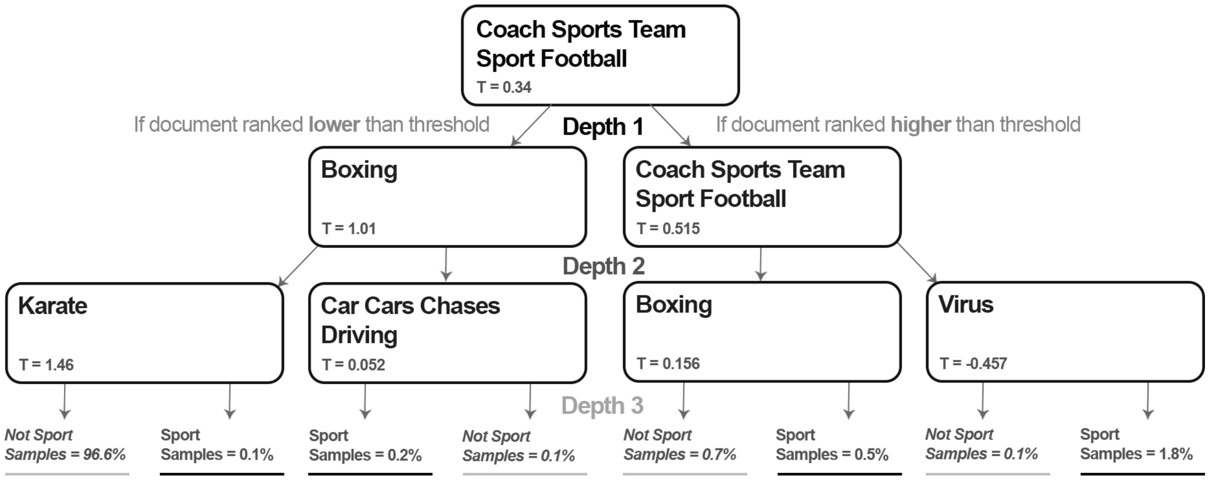

Figure 3.2: An example of a Decision Tree classifying if a movie is in the "Sports" genre. Each Decision Tree Node corresponds to a feature, and the threshold T is equal to the ranking of a document on that feature. The most important direction is used twice, referring to sports and resulting in a majority of negative samples. The nodes at depth three are more specific, sometimes overfitting (e.g. in the case of the "Virus" node) .

features, and are typically labelled with the most probable words according to the corresponding distribution. On the other hand, our work leverages clustering methods to obtain the feature labels. 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 e.g. they allow us to use neural representation learning methods to obtain these spaces. The approach proposed in this Chapter aims to combine the best of both worlds, by providing a way to derive interpretable representations from Vector Space Models. Many extensions of LDA have been proposed to incorporate additional information as well, e.g. aiming to avoid the need to manually specify the number of topics [50], modelling correlations between topics [3], or by incorporating meta-data such as authors or time stamps [44, 59]. Nonetheless, such techniques for extending LDA offer less flexibility than neural network models, e.g. for exploiting numerical attributes or visual features. For comparison, in our experiments the standard topic model algorithm Latent Dirchlet Allocation (LDA) is used as a baseline to compare to the new methodology that transforms standard Vector Space Model representations.

There is much work on learning interpretable representations, with one popular way being to introduce sparsity or non-negativity constraints while learning, for example, sparse PCA learned using the l1-norm, [21] [61], or Non-Negative Sparse Embeddings (NNSE) [40] which are sparse interpretable word-vectors obtained using sparse-matrix factorization and non-negativity constraints. A similar technique can also be applied to distributional word-embeddings by integrating this method with the Skip-Gram model [36]. However, our approach is not intended to transform the learning processes, but rather be a post-processing step on an existing representation.

Similar to the approach in this chapter, [15] introduce a post-processing method to convert any distributional word-vector into sparse word vectors, which additionally satisfy our idea of disentangled interpretability. However, the representation produced by the method in this work differs from sparse representations in that it is dense, where each feature is salient and interpretable. Another method is to describe a representation, e.g. sense word-embeddings that are linked to synsets [42] in-order to make them interpretable. Although this is a post-processing step similar to our method, this is a linking rather than a transformation of the representation.

Another method is to integrate grammatical structure into the learning of the representation, for example [33] obtained a representation learned with attention mechanisms on the dependency structures of sentences, but this differs from the intention of our work, which is not to introduce new structures to the representation to make it more interpretable but instead use the already existing structure to obtain an interpretable representation. For short interpretable documents, [37] introduced tax2vec, which produced interpretable features from word taxonomies, useful for low data models. In [**?**] word-vectors were clustered and then used as a bag-of-clusters, where if a word occurs in those word-vector clusters it contributes to the Bag-Of-Words frequency. Although clustering is used in the method, it is not used to create a Bag-Of-Words, instead relying on the spatial relationships in the space as our representation.

This chapter continues as follows: First the method is described, making explicit the variations from to the original method in [11]. This is followed by a qualitative and quantitative analysis, finishing with a conclusion on the benefits and limitations of this approach.

3.2 Method

This section details the methodology to obtain an interpretable representation from only a Vector Space Model and its associated Bag-Of-Words 2.2.4.

3.2.1 Obtaining Directions and Rankings From Words

We explain the method in terms of document classification. Assuming a Bag-Of-Words B_w has an associated vocabulary W_w, in this section we introduce the first step: how to obtain feature-directions D_w for each word W_w, and rankings of documents on these directions r_w, where each word is ranked on every document. For this step, not all words are expected to be salient in the domain. Instead, the first step shows how to obtain an interpretable representation where every document is ranked on every word, and the next step shows how to filter these rankings to only salient features.

Obtaining directions for each word Each document is referred to as a point d_p in the vector space model S_d. For each word w, a hyper-plane is obtained h_w from a Linear Support Vector Machine (See Section 2.2.2[1]) that is trained on the Bag-Of-Words representation so that document points d_p in the space V_d where the word w occurred more than once for that document $d_{wf} >= 1$ are separated from those where the word did not occur $d_{wf} == 0$. This process is repeated such that a hyper-plane is found for all words in the vocabulary above a frequency threshold $w_f > T$ where T is chosen such that words which are infrequent enough to cause the classifier to overfit are not included. As this task is unbalanced, i.e. there are typically less documents that contain the word compared to those that do not contain it, the weights of the classifier are balanced such that positive instances are weighted in proportion to how rare they are.

As previously mentioned, not all words will be influential on the structure of the representation. Only words that are salient will be well separated. Although the hyperplane h_w learned is binary (either classifying documents d_p as negative or positive), it can be expected that the distance of the document points d_p from the hyperplane boundary varies, as the space's V_w similarity structure is in degrees rather than hard boundaries. For example in a space constructed from

[1]This was also tested using a logistic regression classifier, and achieved similar results

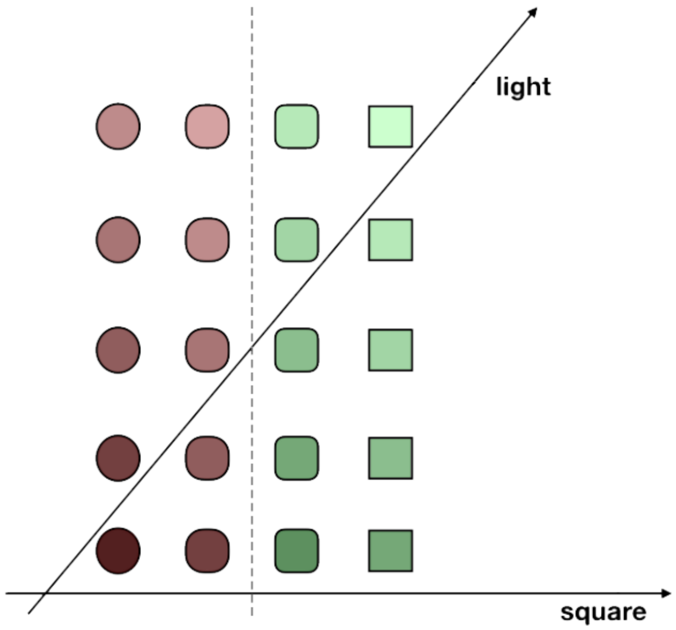

Figure 3.3: An example of a hyper-plane and its orthogonal direction in a toy domain of shapes. Green shapes are positive examples and red shapes are negative examples, but despite the problem being binary those closest to the hyper-plane are less defined than those further away, resulting in the orthogonal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

frequency vectors W_{wf}, it can be expected that the documents which contain the word more frequently would be further away from the hyper-plane on the positive side. Similarly, in the case of our experiments, the documents with a higher PPMI value will be more distant from the hyper plane on the positive side. This is the insight that informs the method to obtain the direction.

The vector v_w perpendicular to the hyperplane h_w is taken as a direction D_w that models documents d_p from the lowest document ranked on the word w (at the distance furthest from the hyperplane on the side where documents d_p are classified) to the highest ranked on the word w at the distance furthest from the hyperplane at the positive side. An example of such a direction D_w is shown in the toy domain in Figure 3.3. To apply this idea to a real domain, we can give an example from movie reviews, where the word is 'Scary' and the most 'Scary' movies are at the tip of the direction and those that are least 'Scary' are at the base of the direction.

Ranking documents on directions Although we do refer to the direction D_w ranking documents on a word w, we do not yet have a specific value to represent this ranking. Once a feature-direction vector is obtained for each word D_w the next step is to quantify the degree to which each document d_p has that word, by obtaining a value that corresponds to how far-up it is on the direction vector D_w. If d_p is the representation of a document in the given vector space as a point then the dot product is used between the direction vector for the word D_w and the document vector $D_w \cdot p_d$ as the ranking r_{dw} of the document d for the word w, and in particular, we take $r_{d1} < r_{d2}$ to mean that the document d_2 'has' the feature to a greater extent than d_1 (e.g. in a domain of movie reviews if the word is 'cinematography', the movie will likely have notable cinematography). Once the dot product value is obtained in this way for each document on a word, this forms a ranking feature for the word. By obtaining a ranking of all documents on all words, a rankings matrix of size $d_n * w_n$ can be obtained. This representation forms the basis of the method, with future steps removing those directions that are not salient, and then clustering similar directions together.

3.2.2 Filtering Words

In this section, words are filtered that do not influence the structure of the domain. This is done by evaluating them using a scoring metric, and removing the words that are not sufficiently well scored. Originally, only binary features were considered. These binary features were measured in terms of the performance of the SVM classifier. If the hyperplane correctly separates the entities well, 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 Vector Space Model representation of d. However, this approach does not consider the quality of the ranking. To consider this, the new metric Normalized Discounted Cumulative Gain was introduced, using the bag-of-words as its target ranking under the assumption that if a ranking matches the ordered score of a PPMI BOW, then it is a good ranking. If this is the case, it can also be assumed that this means the word was strongly influential in the space, as it retains the detail of the Bag-Of-Words information in the space's structure.

Cohen's Kappa. This is the metric used in the work that originally introduced this method [11]. This is a binary feature evaluation metric that deals with the problem that these words 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, they proposed to use Cohen's Kappa score instead. In our experiments, however, it was found that this can be too restrictive, allowing us to sometimes obtain better results with the more simple accuracy metric.

Classification accuracy. If a model has high accuracy for a word w, it seems reasonable to assume that w describes a salient property for the given domain. However, despite balancing the weights of the original SVM used to obtain the hyper-plane, the value this metric places on correctly predicting negative classification often results in noise particular to this metric being identified, e.g. metadata like a reviewers name that only occurs in a few reviews being given a high accuracy score as the method, as it overfit to only predict negative instances.

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 [24]. It favours initial documents over later ones. Some alternative metrics were tried that did not prioritize the top rankings being correct more, but this came with two problems. First, PPMI has a large number of zero scores. This makes the lower dot product documents have an uneven comparison, disrupting the score based on them being given a non-zero ranking score by the method. The second is that the documents without many occurrences of the word are less prioritized in the space, and largely influenced by other words, making their ranking less reliable. 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, a simple cut-off is applied (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, so in practice this value is taken as a hyper-parameter.

In principal, NDCG should be better suited for gradual features. In practice, however, there was not such a clear pattern in the differences between the words chosen by these metrics despite often finding different words. Put another way, it is difficult to say if the words highly scored by NDCG are more gradual than other scoring metrics.

3.2.3 Labelling Words

Although the rankings of single words are informative for models, it is difficult for a human to grasp the meaning of a word without context. This can be resolved simply by finding the n most similar directions to each word's direction.

Another approach is to use a clustering method like k-means. For these clustering method, the aim is to go from single word directions D_w to clusters of these single word directions C_d labelled by the words clustered together C_w. If we consider two directions, "Blood" and "Gore", both of these are approximating a similar feature of movies, they both relate to how much blood a movie contains. Because of this, their directions will be very similar to each other. This is the first idea behind using a clustering method on these directions. It resolves the issue of repetition in the directions, and if the clustered directions are averaged then that clustered direction will balance between documents that used the word 'Bloody' to describe the bloodiness of the movie and the word 'Gore'. Some films may be 'Bloody', but may not necessarily have the term 'Gore' in their reviews, and vice versa. Or, a review may favour one term over the other. By using a clustering method, a direction could be obtained that more accurately represents the semantics of a bloody film more than either of the terms individually.

It is not always the case that this new clustered direction will perform better than a single relevant direction for a class. In fact, its possible that when clustering many terms together, the ranking can be more disrupted than helped. For example given a cluster $\{Romance, Love\}$ and a cluster $\{Blood, Gore\}$ the direction for $\{Cute\}$ is clearly more relevant to the former rather than the latter, and likely has been used in the reviews for romance movies. But it has also likely been used in reviews for movies containing cute animals. This would make the new clustered direction $\{Romance, Love, Cute\}$ perform worse at classifying the movie genre "Romance", but a bit better at classifying animal movies. Ideally, this feature would form a new cluster - but a balance must be held between retaining the precision of the rankings and introducing new rankings that are appropiately disentangled from the existing ones, without repeating existing concepts. In the quantitative results, sometimes clustering performed worse than single directions, and not being able to find this balance for the specific classes in question can be attributed as to why.

The previous work's clustering method is used, and additionally k-means is experimented with:

Derrac's K-Means Variation This is the clustering method used in the work this method was introduced in [**?**]. As input to the clustering algorithm, it considers 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 each cluster C_j is identified with a set of words. Furthermore v_{C_j} will be written 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 the method.

The first cluster centroid is chosen by taking the top-scoring direction for its associated metric. Then, centroids are selected until the desired amount is reached 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 the centroids are selected, for each remaining direction the centroid is found it is most similar to, and the centroid is updated once the direction has been added.

Cluster centroids are taken as cluster directions, and the representation is obtained by ranking documents on this cluster direction. It is also possible to rank documents on the initial direction only, and use the cluster names as descriptions if the clusters are too noisy.

K-Means Although the previous method does have a method for selecting cluster centres, typically it was found that it relies too much on its initial directions, meaning if a noisy direction is chosen as the first cluster centre, then key directions may be missed. Avoiding this is difficult without extensive and sometimes arbitrary hyper-parameter optimization. For this reason, it was decided to try K-Means as a clustering algorithm. K-means traditionally begins with K centroids c randomly placed into the space. In our case, these centers are weighted according to the squared distance from the closest center already chosen. [1] Then, the distance between each point d_p and centroid c is calculated. In-order for euclidian distance to be meaningful, directions are normalized making euclidian distance the same as cosine similarity. 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

Data Type	Unprocessed	Processed
Newsgroups	morgan and guzman will have era's 1 run higher than last year, and the cubs will be idiots and not pitch harkey as much as hibbard. castillo won't be good (i think he's a stud pitcher)	morgan guzman eras run higher last year cubs idiots pitch harkey much hibbard castillo wont good think hes stud pitcher
Sentiment	All the world's a stage and its people actors in it–or something like that. Who the hell said that theatre stopped at the orchestra pit–or even at the theatre door? Why is not the audience participants in the theatrical experience, including the story itself? This film was a grand experiment that said: "Hey! the story is you and it needs more than your attention, it needs your active participation"". ""Sometimes we bring the story to you, sometimes you have to go to the story."" Alas no one listened, but that does not mean it should not have been said."	worlds stage people actors something like hell said theatre stopped orchestra pit even theatre door audience participants theatrical experience including story film grand experiment said hey story needs attention needs active participation sometimes bring story sometimes go story alas one listened mean said
Reuters	U.K. MONEY MARKET SHORTAGE FORECAST REVISED DOWN The Bank of England said it had revised its forecast of the shortage in the money market down to 450 mln stg before taking account of its morning operations. At noon the bank had estimated the shortfall at 500 mln stg.	uk money market shortage forecast revised bank england said revised forecast shortage money market 450 mln stg taking account morning operations noon bank estimated shortfall 500 mln stg

Table 3.2: Text examples from the first three domains

points assigned to the centroids do not change.

3.3 Qualitative Results

3.3.1 Datasets

The experiments are using five different domains. To begin, the properties of these domains are explained to try to give an insight into the kind of text stored within them. This is to better inform analysis of our qualitative results. Examples are shown in three domains in Table 3.2.

20 Newsgroups[2] Obtained from scikit-learn. [3] Where documents are discussions from one of twenty different groups, specifically Atheism, Computer Graphics, Microsoft Windows, IBM PC Hardware, Mac Hardware, X-Window (GUI Software), Automobiles, Motorcycles, Baseball, Hockey, Cryptography, Electronics, Medicine, Space, Christianity, Guns, The Middle East, General Politics and General Religion. These also act as the classes for the dataset. Originally containing 18,846 documents, in this work it is preprocessed using sklearn to remove headers,

[2] http://qwone.com/ jason/20Newsgroups/

[3] https://scikit-learn.org/0.19/modules/generated/sklearn.datasets.fetch_20newsgroups.html#sklearn.datasets.fetch_20newsgroup

footers and quotes. Then, empty and duplicate documents are removed, resulting in 18302 documents. The vocabulary size (unique words) is 141,321. The data is not shuffled. After filtering out terms that did not occur in at least two documents, ending up with a vocabulary of size 51,064. The number of positive instances averaged across all classes is 942, around 5%.

IMDB Sentiment Obtained from Keras [4] Where documents are IMDB movie reviews, containing 50,000 documents with a vocabulary size of 78588. After removing terms that did not occur in at least two documents, ending up with a vocabulary of size 55384. This is a smaller change than the newsgroups, which began with a larger vocabulary than sentiment, but ended vocabularies about the same. This means that newsgroups contained many terms that were not relevant to a majority of the documents, likely because the 20 different newsgroups spread across so many topics. The corpus is split half and half between positive and negative reviews, with the task being to identify the sentiment of the review, so the number of positive instances in the classes is 25,000.

Reuters-21578, Distribution 1.0 Obtained from NLTK[5]. Documents from the Reuters financial newswire service in 1987, originally containing 10788 documents. After removing empty and duplicate documents, ending up with 10655 documents. It originally contained 90 classes, but as they were extremely unbalanced all classes that did not have at least 100 positive instances were removed, resulting in 21 classes. These classes are Trade, Grain, Natural Gas (nat-gas), Crude Oil (crude), Sugar, Corn, Vegetable Oil (veg-oil), Ship, Coffee, Wheat, Gold, Acquisitions (acq), Interest, Money/Foreign Exchange (money-fx), Soybean, Oilseed, Earnings and Earnings Forecasts (earn), BOP, Gross National Product (gnp), Dollar (dlr) and Money-Supply. The original vocabulary size is 51,0001, and after removing all words that do not occur in at least two documents, the vocabulary size is 22542. The number of positive instances averaged across all classes is 541, around 5%.

Placetypes Taken from work by Derrac [11]. Documents are composed of concatenated flickr tags, where each document, named after a flickr tag, is composed of all flickr tags where that tag occurred. A minimum of 1,000 photos for each tag was a requirement, and the tags selected were taken from three different taxonomies (Geonames, Foursquare and the site category for the common-sense knowledge base OpenCYC). It originally has a vocabulary size of 746,527 and

[4] https://keras.io/datasets/

[5] https://www.nltk.org/book/ch02.html

1383 documents. This is a very large vocabulary size to document ratio. The end vocabulary for this space was 100,000, which is used as a hard limit. This is roughly equivalent to removing all documents that would not be in at least 6 documents. As most classes in this domain are extremely sparse (less than 100 positive instances) no classes are deleted. There are three tasks, generated from three different place type taxonomies. The Foursquare taxonomy, classifying the 9 top-level categories from Foursquare in September 2013, Arts and Entertainment, College and University, Food, Professional and Other Places, Nightlife Spot, Parks And Outdoors, Shops and Service, Travel and Transport and Residence. the GeoNames taxonomy where 7 of 9 categories were used, Stream/Lake, Parks/Area, Road/Railroad, Spot/Building/Farm, Mountain/Hill/Rock, Undersea, and Forest/Heath. The OpenCYC Taxonomy, where 93 categories were used by Derrac, but it was only possible to match 25 of those classes to the representations. As 8 of these remaining classes had a low number of positive occurrences, OpenCYC classes are removed that do not have positive instances for at least 30 documents, leaving us with 17, Aqueduct, Border, Building, Dam, Facility, Foreground, Historical Site, Holy Site, Landmark, Medical Facility, Medical School, Military Place, Monsoon Forest, National Monument, Outdoor Location, Rock Formation, and Room. Naturally as these tasks were derived from taxonomies they are multi-label.

Movies Taken from work by Derrac [11]. A dataset where each document is a movie represented by all of its reviews concatenated across a number of sources (Rotten Tomatoes, IMDB, Amazon Reviews). It starts off with a vocabulary size of 551,080 and a document size of 15,000. However, after investigating the data made available by the authors, it was found that there were a number of duplicate documents. After removing these duplicate documents, there are 13978 documents. In the same way as the place-types, the vocabulary is limited at size 100,000. Three tasks are used to evaluate, 23 movie genres, specifically Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, History, Horror, Music, Musical, Mystery, Romance, Sci-Fi, Short, Sport, Thriller, War, Western. 100 of the most common IMDB plot keywords (See Appendix **??**) and Age Ratings from the UK and US, USA-G, UK-12-12A, UK-15, UK-18, UK-PG, USA-PG-PG13, USA-R.

For each domain, we filter out terms that do not occur in at least two documents, and additionally limit the maximum number of words in a vocabulary to 100,000. For all of these datasets, we split them into a 2/3 training data, 1/3 test data split. We additionally remove the end 20% of the training data and use that as development data for our hyper-parameters, which is then not used for the final models verified using test data. For the movies and place-type domains, the original text was not available.

3.3.2 Space Types

Below the choices for the Vector Space Models that are formally described in Section 2.2.4 are explained:

Multi-Dimensional Scaling (MDS): Following [10], we use Multi-Dimensional Scaling (MDS) to learn semantic spaces from the angular differences between the PPMI weighted BoW vectors.

Principal Component Analysis (PCA): directly uses the PPMI weighted BoW vectors as input, and which avoids the quadratic complexity of the MDS method. A standard dimensionality reduction technique, used as a baseline reference.

Doc2Vec (D2V): Inspired by the Skipgram model [31]. A distributional document representation used as a representative of a higher performing method of learning in terms of document classification. For the Doc2Vec space, the hyper-parameters are additionally tuned for the $windowsize(5, 10, 15)$ referring to the context window, the $mincount(1, 5, 10)$ referring to the minimum frequency of words and the $epochs(50, 100, 200)$ of the network for each size space. The process with our two-part hyperparameter optimization as in this case is as follows: Grid search is used to select the parameters for the representation, then find the most suitable model (e.g. Decision Tree, SVM) for that representation.

Average Word Vectors (AWV): Finally, we also learn semantic spaces by averaging word vectors, using a pre-trained GloVe word embeddings trained on the Wikipedia 2014 + Gigaword 5 corpus[6]. While simply averaging word vectors may seem naive, this was found to be a competitive approach for unsupervised representations in several applications [19]. We simply average the vector representations of the words that appear at least twice in the BoW representation.

[6] https://nlp.stanford.edu/projects/glove/

3.3.3 The best-performing directions for each domain

To give an understanding of the kind-of directions found for each domain, the top-scoring ones are presented in Table 3.3. These are arranged from highest scoring to least scoring, with the score-type and space-type chosen by performance. These are not clusters, but rather single directions with the two most similar directions in brackets beside them for context. This is the alternative way of presenting these directions as mentioned at the start of Section 3.2.3.

There is an interesting difference between the sentiment directions and the movies directions in the examples below. Both of these domains are composed of movie reviews, but the documents in the former are a concatenation of a number of reviews across different sources, while the latter are individual reviews. This has resulted in the more general concepts that apply to many movies being salient in the movies domain, but are less important than the names of actors and actresses in the sentiment domain. This is likely because the PPMI scores for actor names would be high as they are both rare and definitive for movies. For the newsgroups domain, a number of directions are seen that are likely to only belong to a certain newsgroups, e.g. you would find the word 'celestial' more often in the religious sections than the others, and the word 'diesel' more often in the automobile section but not others. This is an expected natural clustering of the domain into its 20 newsgroups. The place-types section generally describes either aspects of the camera (e.g. canon60d), aspects of the photo (greyscale) or features found in the photo (gardening). The former likely relates to the degree to which filters or editing has been applied to the photo, while the latter makes more sense for our classification task. For the reuters dataset, the highest scored semantics seem to generally be related to dates (1st, may, june), however there is also some business jargon (quarterly, avg, dlr).

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)

Table 3.3: The top-scoring words for each domain, scoring metric and space type determined by the highest F1-score

3.3.4 Comparing Space Types

To select these quantitative examples for comparing score types, it was first demonstrated on the movies domain to be consistent with previous examples. However, as this does not contain the doc2vec space, additional results are provided in the next section for the newsgroups. The space that performed well on the genres task for the movies is used, with the understanding that genres as a key natural classification task will likely give good example directions that correspond to domain knowledge. After selecting this space, the same sized spaces are chosen from the other space-types (size 200). The same score-type and frequency cut-off as the best performing space-type are also used. In this case, the best performing type for the PCA space was 20,000 frequency cutoff and NDCG. So even though sometimes a different frequency cut-off performed better for the other space-types, this is equalized so that the words are the same. This means that sometimes the space-type is a slightly worse performing one than chosen as the final results, and that the original space has a performance advantage, but this makes the results more consistent. These qualitative experiments are approached 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. For this reason, the directions which are unique to each space-type are shown.

When examining the table of results, it can be observed that the common terms are mostly salient concepts relevant to the domain. However, MDS has the most unique general concepts relevant to the domain that others do not have. AWV contains names, and concepts which are interesting but more related to specific aspects than genre (train, slaves). Meanwhile PCA seems to prioritize words in the reviews that are not concepts but rather parts of sentences (surprisingly, admit, talents, tired, anymore). However, both PCA and MDS contain unique noisy terms as well. The term 'berardinelli' and 'rhodes' for MDS as well as 'compuserve' for PCA are artifacts of the data being obtained from the web. Despite this, it seems that MDS does contain more interesting unique directions than PCA, and as it performed best on the genres task, this makes sense.

MDS	AWV	PCA	Common
berardinelli *(employers, distributor)*	billy *(thrown, dirty)*	amount *(leaving, pick)*	noir *(fatale, femme)*
crawford *(joan, davis)*	brother *(brothers, boys)*	fails *(fit, pick)*	gay *(homosexual, homosexuality)*
hitchcocks *(hitchcock, alfred)*	fonda *(henry, jane)*	pick *(fails, fit)*	prison *(jail, prisoners)*
warners *(warner, bros)*	building *(built, climax)*	stands *(fails, cover)*	arts *(rec, robomod)*
nuclear *(weapons, soviet)*	train *(tracks, thrown)*	surprisingly *(offer, fit)*	allens *(woody, allen)*
joan *(crawford, barbara)*	slaves *(slavery, excuse)*	copyright *(email, compuserve)*	jokes *(laughs, joke)*
kidnapped *(kidnapping, torture)*		length *(reflect, expressed)*	animation *(animated, cartoon)*
hop *(hip, rap)*		profanity *(reflect, producers)*	sherlock *(holmes, detective)*
kung *(martial, jackie)*		compuserve *(copyright, internetreviews)*	western *(westerns, wayne)*
ballet *(dancers, dancer)*		talents *(admit, agree)*	songs *(song, lyrics)*
gambling *(vegas, las)*		admit *(agree, talents)*	comedies *(comedic, laughs)*
alcoholic *(drunk, alcoholism)*		developed *(introduced, sounds)*	workout *(exercise, challenging)*
waves *(surfing, wave)*		intended *(bother, werent)*	laughs *(funnier, hilarious)*
jaws *(jurassic, godfather)*		constantly *(putting, sounds)*	drug *(drugs, addict)*
jungle *(natives, island)*		tired *(anymore, mediocre)*	sci *(science, fiction)*
employers *(berardinelli, distributor)*		produced *(spoiler, surprising)*	documentary *(documentaries, interviews)*
pot *(weed, stoned)*		involving *(believes, belief)*	students *(student, schools)*
canadian *(invasion, cheap)*		anymore *(continue, tired)*	thriller *(thrillers, suspense)*
murphy *(eddie, comedian)*		leaving *(fit, pick)*	allen *(woody, allens)*
comics *(comedian, comedians)*		makers *(producers, aspects)*	funniest *(hilarious, laughing)*
kidnapping *(kidnapped, torture)*		introduced *(developed, considered)*	gags *(jokes, slapstick)*
subscribe *(email, internetreviews)*		loses *(climax, suffers)*	adults *(children, adult)*
vegas *(las, gambling)*		negative *(positive, bother)*	animated *(animation, cartoon)*
distributor *(berardinelli, employers)*		expressed *(reflect, opinions)*	dancing *(dance, dances)*
wave *(waves, surfing)*		mildly *(mediocre, forgettable)*	teen *(teenage, teens)*
rhodes *(internetreviews, email)*		helped *(putting, allowed)*	soldiers *(soldier, army)*
hippie *(pot, sixties)*		reflect *(expressed, opinions)*	indie *(independent, festival)*
weed *(pot, stoned)*		opinions *(reflect, expressed)*	suspense *(suspenseful, thriller)*
caribbean *(pirates, island)*		frequently *(occasionally, consistently)*	creepy *(scary, eerie)*
eddie *(murphy, comedian)*		content *(agree, proves)*	italian *(italy, spaghetti)*
sixties *(beatles, hippie)*		allowed *(helped, werent)*	jews *(jewish, nazis)*
... 8 More		suffers *(lacks, loses)*	... 1480 more

Table 3.4: Unique terms between space-types

Score Types

There are unique directions for each different space type from the movies domain, each suitable to different tasks. Obtained in the same way as before, this time the 200 MDS space is used that performed the best on the genres task and found those unique to it. Once again, the most understandable and general concepts are those that are common to all score-types. NDCG performed the best on most tasks, and it can be seen that a lot of new concepts are introduced in NDCG compared to the other scoring types. F1 by and large seems is difficult to understand, referring to names or specific aspects of the scene, and accuracy is similar. Kappa has some unique sentiment related terms, as as well as some aspects of the presentation of the film (featurette, critic, technical), but it does not contain unique general concepts the way NDCG does. It can be surmised that as NDCG contains these unique conceptual directions, it is able to perform better than other score-types.

NDCG	**F1**	**Accuracy**	**Kappa**	**Common**
gay *(homosexuality, sexuality)*	company *(sell, pay)*	kennedy *(republic, elected)*	definately *(alot, awesome)*	horror *(scares, scares)*
arts *(hong, chan)*	street *(city, york)*	bags *(listened, salvation)*	guns *(gun, shoot)*	laughs *(funnier, funnier)*
sports *(win, players)*	red *(numerous, fashion)*	summers *(verge, medieval)*	flawless *(perfection, brilliantly)*	jokes *(gags, gags)*
apes *(remembered, planet)*	project *(creating, spent)*	revolve *(sincerely, historian)*	mail *(reviewed, rated)*	comedies *(comedic, comedic)*
german *(germans, europe)*	mark *(favor, pull)*	locale *(foster, sharply)*	garbage *(crap, horrible)*	sci *(scifi, alien)*
satire *(parody, parodies)*	lady *(actress, lovely)*	cooler *(downward, reports)*	featurette *(featurettes, extras)*	funniest *(hilarious, hilarious)*
band *(rock, vocals)*	fire *(ground, force)*	spades *(ralph, medieval)*	complaint *(extra, added)*	creepy *(spooky, spooky)*
crude *(offensive, offended)*	post *(essentially, purpose)*	filmography *(ralph, experiments)*	mission *(enemy, saving)*	thriller *(thrillers, thrillers)*
dancing *(dance, dances)*	heads *(large, throw)*	quentin *(downward, anime)*	ruin *(wondering, heck)*	funnier *(laughs, laughs)*
restored *(print, remastered)*	water *(land, large)*	employers *(finishes, downward)*	wars *(forces, enemy)*	suspense *(suspenseful, suspenseful)*
drugs *(drug, abuse)*	road *(drive, trip)*	formal *(victory, kennedy)*	prefer *(compare, added)*	gore *(gory, gory)*
church *(religious, jesus)*	brother *(son, dad)*	tube *(esta, muscle)*	heroes *(packed, hero)*	gags *(jokes, jokes)*
sexuality *(sexual, sexually)*	party *(decide, hot)*	woefully *(restless, knockout)*	necessarily *(offer, draw)*	science *(sci, sci)*
sexually *(sexual, sexuality)*	badly *(awful, poorly)*	scientists *(hilarity, locale)*	portray *(portrayed, portraying)*	gory *(gore, gore)*
england *(british, english)*	limited *(aspect, unlike)*	overboard *(civilized, cinderella)*	critic *(reviewed, net)*	government *(political, political)*
ocean *(sea, boat)*	impression *(instance, reasons)*	rumors *(homosexuality, characteristics)*	reviewed *(rated, mail)*	suspenseful *(suspense, suspense)*
marry *(married, marriage)*	trip *(journey, road)*	salvation *(bags, cooler)*	saving *(carry, forced)*	frightening *(terrifying, terrifying)*
campy *(cult, cheesy)*	michael *(producers, david)*	actively *(assassination, overcoming)*	technical *(digital, presentation)*	military *(army, army)*
christian *(religious, jesus)*	memory *(forgotten, memories)*	stretching *(victory, hideous)*	statement *(exist, critical)*	slapstick *(gags, gags)*
melodrama *(dramatic, tragedy)*	james *(robert, michael)*	downward *(cooler, crawling)*	shocked *(hate, warning)*	scary *(scare, scare)*
sing *(singing, sings)*	thin *(barely, flat)*	rocked *(staple, demented)*	flying *(air, force)*	blu *(unanswered, ray)*
sentimental *(touching, sappy)*	pre *(popular, include)*	affectionate *(esta, muscle)*	danger *(dangerous, edge)*	internetreviews *(rhodes, rhodes)*
depressing *(bleak, suffering)*	faces *(constant, unlike)*	protest *(protective, assassination)*		cgi *(computer, computer)*
evidence *(investigation, accused)*	values *(exception, wise)*	confined *(cooler, downward)*		email *(web, web)*
adorable *(cute, sweet)*	unusual *(odd, seemingly)*	inhabit *(quentin, drawback)*		thrilling *(thrill, exciting)*
episodes *(episode, television)*	lovers *(lover, lovely)*	latin *(communities, mount)*		web *(email, email)*
teenager *(teen, teenage)*	frame *(image, effect)*	reception *(como, finishes)*		horror *(scares, scares)*
magical *(fantasy, lovely)*	mans *(ultimate, sees)*	uptight *(suspensful, stalked)*		laughs *(funnier, funnier)*
health *(medical, suffering)*	efforts *(generally, nonetheless)*	brink *(inexplicable, freddy)*		suspense *(suspenseful, suspenseful)*

Table 3.5: Different score types

Comparing PPMI representations to doc2vec

Now in Table a comparison is shown between a time when doc2vec was the highest performing representation, in this case on the newsgroups domain. Doc2vec is compared to MDS in this case as MDS also performed well. This is to see if doc2vec, by making use of word-vectors and word-context can find interesting unique directions compared to MDS, which was obtained from a PPMI BOW. In general, it is found that MDS contains a lot more irrelevant words than D2V, specifically related to parts-of-words. It seems that doc2vec was better at recognizing these words as noise and uninteresting compared to PPMI, which must have prioritized these words. Doc2Vec also brings up some interesting concepts, e.g. cryptology, which is very relevant to the 20 newsgroup subtype of cryptography. It can be expected that by using word vectors, doc2vec is able to more easily identify interesting words and de-prioritize words which are common to the english language despite potentially being more rare in a smaller dataset.

3.4 Quantitative Results

3.4.1 Evaluation Method

Primarily the effectiveness of a representation is evaluated on its ability to perform in low-depth Decision Trees, specifically CART Decision Trees (See Background Section 2.2.1) with a limited depth of one, two and three. This evaluation has a few assumptions: A good interpretable representation disentangles salient domain knowledge into its dimensions, and natural domain tasks (e.g. classifying genres of movies using their reviews) can be evaluated effectively using that salient domain knowledge. Put another way, if the space is representing domain knowledge well it can be expected that the space is linearly separable for key semantics of the domain. In spatial terms, a representation will be capable of being linearly transformed by our method into these distinct relevant concepts if semantically distinct entities are spatially separated, and semantically similar entities are close together.

If only the the quality of the representation was being evaluated, only Linear SVM's could be used to find the hyper-planes that effectively separate these spatial representations for the class. However, the representations that encode this spatial information are not interpretable,

D2V	MDS	Common
leftover (pizza, brake)	hi (folks, everyone)	chastity (shameful, soon)
wk (5173552178, 18084tmibmclmsuedu)	looking (spend, rather)	n3jxp (gordon, gebcadredslpittedu)
eng (padres, makefile)	need (needs, means)	skepticism (gebcadredslpittedu, n3jxp)
porsche (nanao, 1280x1024)	post (summary, net)	anyone (knows, else)
diesel (cylinders, steam)	find (couldnt, look)	gebcadredslpittedu (soon, gordon)
scorer (gilmour, lindros)	hello (kind, thank)	intellect (soon, gordon)
parliament (caucasus, semifinals)	david (yet, man)	please (respond, reply)
atm (padres, inflatable)	got (mine, youve)	thanks (responses, advance)
cryptology (attendees, bait)	go (take, lets)	email (via, address)
intake (calcium, mellon)	question (answer, answered)	know (let, far)
433 (366, 313)	interested (including, products)	get (wait, trying)
ghetto (warsaw, gaza)	list (mailing, send)	think (important, level)
lens (lenses, ankara)	sorry (guess, hear)	good (luck, bad)
rushdie (sinless, wiretaps)	heard (ever, anything)	shafer (dryden, nasa)
immaculate (porsche, alice)	cheers (kent, instead)	bobbeviceicotekcom (manhattan, beauchaine)
keenan (lindros, bosnian)	say (nothing, anything)	dryden (shafer, nasa)
boxer (jets, hawks)	number (call, numbers)	im (sure, working)
linden (mogilny, 176)	mailing (list, send)	sank (bronx, away)
candida (yeast, noring)	call (number, phone)	banks (soon, gordon)
octopus (web, 347)	thank (thanx, better)	like (sounds, looks)
czech (detectors, kuwait)	read (reading, group)	shameful (soon, gordon)
survivor (warsaw, croats)	phone (company, number)	could (away, bobbeviceicotekcom)
5173552178 (circumference, wk)	mail (send, list)	would (appreciate, wouldnt)
18084tmibmclmsuedu (circumference, wk)	doesnt (isnt, mean)	beauchaine (bobbeviceicotekcom, away)
3369591 (circumference, wk)	lot (big, little)	ive (seen, never)
mcwilliams (circumference, wk)	thats (unless, youre)	surrender (soon, gebcadredslpittedu)
coldblooded (dictatorship, czech)	believe (actually, truth)	problem (problems, fix)
militia (federalist, occupying)	youre (unless, theyre)	windows (31, dos)
cbc (ahl, somalia)	send (mail, mailing)	gordon (soon, gebcadredslpittedu)

Table 3.6: Comparing an MDS sapce to a D2V space for Newsgroups, where a D2V space performed best..
3.3.4

so a linear classifier although able to separate the documents that contain the class and do not contain them will not be interpretable either. It is our main interest to evaluate how well a representation encodes these key semantics while also being restricted by the requirement to be disentangled into words or clusters, in other words how well it represents the information while also being interpretable.

Given these assumptions, low-depth Decision Trees can give an estimation of how good an interpretable representation is. If the representation cannot perform for a class at a one-depth tree, then it is not disentangled such that it contains a single salient dimension that effectively evaluates a class. If a representation cannot perform well on two-depth trees, then the representation is not disentangled into three concepts that can sufficiently determine that class, and if a representation cannot perform well on three-depth trees, it has not disentangled the representation such that there are nine relevant concepts that are relevant to that class. To see what these different trees look like see Figure **??**. A comparison to put this in better perspective is to an unbounded tree. Unbounded trees select a large amount of dimensions in order to achieve a performance difference on development data, but when applied to test data the models do not generalize well. This is because they overfit, rather than using the key semantics of the space to classify.

Primarily F1-score is used to determine if a classifier is good or not. This is because many of the classes are unbalanced so accuracy is not a good metric, as high accuracy could be achieved by predicting only zeros. All of the results shown in this section are the end-product of a two-part hyper-parameter optimization. Each Decision Tree has its own set of hyper-parameters that are optimized as does each representation-type. These are the models trained on the training data and scored on the test data, with the highest performing in terms of F1-score parameters from hyper-parameter optimization on the development data. For ease of comparison, some results are provided with SVM's and unbounded Decision Trees, as well as a baseline Topic Model, which is used as a reference for a standard interpretable representation. Below, the parameters are listed that are optimized for each of these model types:

Linear Support Vector Machines (SVM's)2.2.2: C parameters and gamma parameters. C 1.0, 0.01, 0.001, 0.0001, Gamma 1.0, 0.01, 0.001, 0.0001.

Topic Models2.3: Two priors: The doc topic prior 0.001, 0.01, 0.1 and the topic word prior 0.001, 0.01, 0.1

CART Decision Trees 2.2.1: The number of features to consider when looking for the best split. $None, auto, log2$ and the criterion for a node split $criterion : gini, entropy$.

For the baselines, four different Vector Space Models are used, a Bag-Of-Words of PPMI (BOW-PPMI) scores and a standard Latent Dirchlet Allocation (LDA) Topic Model. As well as the original filtering done to the representations, for the BOW-PPMI additionally all terms are filtered out that do not occur in at least $(d_N/1000)$ documents. Otherwise, there would be too many irrelevant terms to be a fair comparison. The dimension amounts that are compared are of size $(50, 100, 200)$. The MDS space is not available for sentiment, as the memory cost was too prohibitive with 50,000 documents, and there are no doc2vec spaces for placetypes/movies, as it was only possible access to the Bag-Of-Words representation.

When obtaining the single word directions, starting with all of the baseline representations and vocabularies, the infrequent terms are filtered from these vocabularies according to a hyper-parameter that is tuned. As the doc2vec has already been hyper-parameter optimized, the optimal doc2vec space that scored the highest for its class on a Linear SVM is used, rather than tuning the entire process around the doc2vecs vectors. So for example, when evaluating the Keywords task for the movies, directions are obtained from the doc2vec space that performed best for a linear SVM on the Keywords task following the previous experiments.

Results are obtained for the rankings induced from these word directions on Decision Tree's limited to a depth-three in-order to select the best parameters when using directions for each class. The parameters that are desirable to determine are the type of Vector Space Model, the size of the space, the frequency threshold and the score threshold, which determines the top scoring directions. To do so, for each space-type of each size, a grid search is used to find the best frequency and score cut-offs for that sized space-type. Then, from these space-types and sizes the best performing one is selected. There is 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 to 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 Vector Space Model and Scoring-type for that class. 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, classifiers performed better with more data, so we use 20000 as our frequency cutoff and 2000 as our score cutoff. Our hyper-parameters for the frequency cut-off were 5000, 10000 and 20000, and our hyper-parameters for the score-cutoff were 1000 and 2000.

We continue with the optimal space and score-type chosen by the single direction experiments, and use the same frequency and score thresholds as before. Two different clustering algorithms are experimented with: 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 infrequent 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.

For K-means clustering, we use Mini batch K-means, implemented by scikit-learn [7], introduced by [48] and kmeans++ to initialize [1]

3.4.2 Summary of all Results

To begin, the original dimensions of the space are compared to the rankings on single words, the rankings on cluster directions, and the Bag-Of-Words of PPMI scores and topic models on low-depth Decision Trees. Single directions or clusters outperform the baselines in most cases, with the exceptions being in the place-types domain and the keywords task for the movies. For the keywords task, the natural explanation is that in a depth-1 tree, finding words which are directly corresponding to particular keywords is easier with words than if using directions, not only because certain words may have been filtered out, but also because as they are infrequent they may not be well-represented in the space. In this case, the PPMI representation is perfect, as it can find 1-1 matches with the classes without the representations of those words being spatially influenced by other similar words, as it can be expected for them to be in the space. However, this changes when going from depth-one to depth-two and depth-three, which is likely due to overfitting in the case of the PPMI representation. Sometimes Decision Trees of depth-two outperform those of depth-one, but generally depth-three trees perform best. In the case of the place-types, although topic models and PPMI representations are indeed the best, it is not

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

by a wide-margin. Meanwhile when the single directions perform the best in these domains for other tree types they perform much better than the other approaches. Additionally, place-types is our most unbalanced domain with the least documents, so it is possible that they overfit.

	Genres			Keywords			Ratings		
Movies	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	0.23	**0.233**	**0.224**	0.466	0.499	0.498
Clusters	0.431	**0.513**	**0.506**	0.215	0.22	0.219	**0.504**	**0.507**	**0.513**
PPMI	0.429	0.443	0.483	**0.243**	0.224	0.224	0.47	0.453	0.453
Topic	0.415	0.472	0.455	0.189	0.05	0.075	0.473	0.243	0.38
	Newsgroups			Sentiment			Reuters		
	D1	D2	D3	D1	D2	D3	D1	D2	D3
Rep	0.251	0.366	0.356	0.705	0.77	0.773	0.328	0.413	0.501
Single dir	0.418	**0.49**	**0.537**	0.784	0.814	**0.821**	**0.678**	**0.706**	0.72
Cluster	0.394	0.433	0.513	0.735	**0.844**	0.813	0.456	0.569	0.583
PPMI	0.33	0.407	0.444	0.7	0.719	0.73	0.616	0.699	**0.723**
Topic	**0.431**	0.423	0.444	**0.79**	0.791	0.811	0.411	0.527	0.536
	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	0.34	0.367
Single dir	**0.541**	0.498	**0.531**	0.404	**0.428**	0.39	**0.444**	**0.533**	**0.473**
Cluster	0.462	0.507	0.496	**0.413**	0.42	**0.429**	0.444	0.458	0.47
PPMI	0.473	**0.512**	0.491	0.371	0.351	0.352	0.361	0.301	0.242
Topic	0.488	0.433	0.526	0.365	0.271	0.313	0.365	0.3	0.219

Table 3.7: summary of all results

3.4.3 Baseline Representations

In Table 3.8 all variations of the baseline representations used directly as input to Decision Trees and SVM's are shown. These examples that do not apply our methodology, serve as a reference point for what is possible using standard linear models without the need for interpretability. In the representations, there is a big performance drop when going from depth three trees to depth one trees. These kind of performance drops are expected for these representations, as they do not have dimensions that correspond to key semantics, so it is unlikely that a smaller tree can use the available dimensions to model a class with limited depth. In this full table the precision and recall scores are included for clarity, mainly to explain why the high recall scores occur. This is because the weights are balanced as a hyper-parameters, and when the weight is balanced so that positive instances are weighted more heavily, the model prioritizes recall over precision. When this high recall score doesn't occur, that means that not balancing the weights performed better on the development data.

The size of the space is not as influential as the representation type in these results for the Decision Trees. For this reason only the best performing representation of each type are shown in Table 3.8. Out of the space-types, PCA performed much better than its counterparts for reuters, newsgroups and sentiment. The MDS representation performs comparably well using a unrestricted depth tree or an SVM, which shows that with a classifier that can make use of all the dimensions, the performance does not decrease as much. This is likely due to the way that PCA orders its dimensions in importance, resulting in key semantics in its first dimensions, giving it an advantage in low-depth Decision Trees. However, this does not necessarily mean that it contains better directions. In the single directions results, PCA is outperformed by MDS and other representations in F1 score for low Decision Tree depths in any of these domains, with the exception of the depth-two trees for sentiment. Despite MDS often encoding the key semantics across more dimensions than other representations, our method is still able find meaningful directions from this space. There is little link between performance on the raw dimensions of the space and performance with rankings on directions in low-depth Decision Trees. This is somewhat counterintuitive, as it would be normal to expect that a representation which performs poorly when used directly as input to a classifier would have similar performance after a linear transformation, but the reason that it works in our case is because low-depth Decision Trees rely on key semantics being disentangled into individual dimensions. Despite the information encoded in the space, if it is not disentangled then the classifier will not perform well.

Newsgroups	D1				D2				D3				DN				SVM			
	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec
PCA 200	0.701	0.251	0.148	0.811	0.843	0.366	0.245	0.719	0.956	0.355	0.54	0.265	0.946	0.44	0.45	0.432	0.969	0.612	0.746	0.519
PCA 100	0.698	0.247	0.146	0.813	0.835	0.362	0.241	0.731	0.957	0.356	0.576	0.257	0.948	0.451	0.465	0.438	0.969	0.586	0.768	0.474
PCA 50	0.68	0.24	0.141	0.829	0.834	0.355	0.234	0.735	0.957	0.329	0.472	0.253	0.947	0.45	0.462	0.438	0.966	0.52	0.745	0.399
AWV 200	0.687	0.217	0.126	0.781	0.758	0.256	0.156	0.718	0.764	0.26	0.157	0.751	0.937	0.339	0.352	0.328	0.961	0.468	0.641	0.369
AWV 100	0.677	0.21	0.122	0.775	0.78	0.275	0.173	0.683	0.746	0.25	0.149	0.769	0.934	0.324	0.332	0.317	0.865	0.4	0.265	0.812
AWV 50	0.696	0.219	0.127	0.772	0.777	0.272	0.168	0.71	0.743	0.25	0.149	0.786	0.935	0.325	0.335	0.316	0.842	0.362	0.233	0.819
MDS 200	0.581	0.184	0.103	**0.837**	0.742	0.262	0.16	0.729	0.719	0.236	0.139	0.785	0.935	0.327	0.332	0.323	0.965	0.501	**0.802**	0.364
MDS 100	0.586	0.187	0.105	0.833	0.754	0.261	0.159	0.727	0.705	0.236	0.138	**0.808**	0.935	0.33	0.338	0.321	0.878	0.439	0.308	0.765
MDS 50	0.593	0.153	0.087	0.647	0.716	0.25	0.15	**0.756**	0.736	0.243	0.144	0.774	0.935	0.324	0.335	0.313	0.854	0.394	0.259	0.821
D2V 200	0.682	0.205	0.119	0.746	0.802	0.268	0.169	0.646	0.77	0.269	0.164	0.75	0.94	0.366	0.389	0.346	0.961	0.468	0.641	0.369
D2V 100	0.682	0.208	0.12	0.762	0.792	0.268	0.168	0.662	0.786	0.268	0.164	0.727	0.94	0.376	0.392	0.361	**0.971**	**0.628**	0.761	0.535
D2V 50	0.683	0.207	0.12	0.764	0.809	0.294	0.187	0.694	0.782	0.28	0.172	0.761	0.943	0.394	0.415	0.376	0.97	0.601	0.758	0.497
PPMI	**0.948**	0.33	**0.532**	0.239	0.947	0.407	0.511	0.338	0.944	**0.444**	0.506	0.396	**0.951**	**0.494**	**0.496**	**0.492**	0.962	0.613	0.627	0.599
Topic	0.852	**0.431**	0.304	0.743	**0.96**	**0.423**	**0.604**	0.326	**0.961**	0.444	**0.606**	0.35	0.944	0.432	0.434	0.429	0.879	0.46	0.318	**0.835**

Table 3.8: Full results for the newsgroups.

Table 3.9: Results for all other domains for the representations.

Reuters	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.847	0.328	0.917	0.413	0.978	0.501	0.978	0.565	0.989	0.761
AWV	0.782	0.252	0.971	0.328	0.974	0.417	0.973	0.495	0.987	0.719
MDS	0.791	0.263	0.9	0.357	0.979	0.489	0.976	0.522	0.988	0.67
D2V	0.818	0.268	0.867	0.298	0.974	0.445	0.971	0.482	0.986	0.724
PPMI	**0.975**	**0.616**	**0.978**	**0.699**	**0.98**	**0.723**	**0.984**	**0.746**	**0.99**	**0.8**
Topic	0.92	0.411	0.977	0.527	0.977	0.536	0.977	0.56	0.95	0.513

Sentiment	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.745	0.705	0.755	0.77	0.778	0.773	**0.781**	**0.779**	**0.891**	**0.893**
AWV	0.642	0.652	0.643	0.694	0.695	0.717	0.66	0.663	0.827	0.829
D2V	0.642	0.664	0.66	0.707	0.702	0.7	0.711	0.708	0.878	0.878
PPMI	0.616	0.7	0.655	0.719	0.675	0.73	0.712	0.71	0.887	0.888
Topic	**0.793**	**0.79**	**0.794**	**0.791**	**0.81**	**0.811**	0.733	0.73	0.815	0.822

Placetypes OpenCYC	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.586	0.346	0.708	0.343	0.695	0.342	0.832	0.309	0.847	0.474
AWV	0.625	**0.383**	0.651	0.376	0.728	**0.396**	**0.844**	**0.362**	0.85	0.466
MDS	0.624	0.364	0.7	**0.397**	0.731	0.374	0.843	0.305	0.861	**0.476**
PPMI	**0.728**	0.371	0.75	0.351	0.739	0.352	0.843	0.323	**0.9**	0.366
Topic	0.708	0.365	**0.87**	0.271	**0.87**	0.313	0.831	0.313	0.808	0.407

Movies Genres	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.722	0.301	0.755	0.339	0.717	0.321	0.884	0.372	**0.925**	0.518
AWV	0.679	0.29	0.774	0.321	0.756	0.343	0.873	0.312	0.922	0.496
MDS	0.679	0.298	0.79	0.358	0.773	0.354	0.887	0.385	0.875	**0.532**
PPMI	**0.852**	**0.429**	**0.91**	0.443	**0.912**	**0.483**	0.882	**0.416**	0.923	0.526
Topic	0.767	0.415	0.905	**0.472**	0.912	0.455	**0.889**	0.415	0.843	0.491

Placetypes Foursquare	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.731	0.342	0.823	0.393	0.86	0.388	0.887	0.398	0.896	0.568
AWV	0.767	0.401	0.828	0.478	0.85	0.452	0.905	**0.505**	0.923	**0.622**
MDS	**0.915**	0.438	0.804	0.427	0.86	0.454	0.893	0.462	0.932	0.619
PPMI	0.889	0.473	0.915	**0.512**	0.904	0.491	0.881	0.31	**0.938**	0.567
Topic	0.864	**0.488**	**0.916**	0.433	**0.917**	**0.526**	**0.907**	0.464	0.916	0.569

Movies Keywords	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.647	0.185	0.644	0.193	0.677	0.199	0.846	0.161	0.787	0.272
AWV	0.5	0.16	0.641	0.179	0.595	0.174	0.853	0.141	0.717	0.23
MDS	0.633	0.179	0.69	0.198	0.674	0.201	0.84	0.163	0.788	**0.28**
PPMI	**0.818**	**0.243**	0.745	**0.224**	0.739	**0.224**	0.847	**0.17**	**0.921**	0.217
Topic	0.629	0.189	**0.932**	0.05	**0.93**	0.075	**0.857**	0.152	0.678	0.21

Placetypes Geonames	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	0.502	0.301	0.69	0.305	0.68	0.295	0.821	0.243	0.844	0.401
AWV	0.657	0.326	0.755	0.323	0.842	**0.367**	0.813	0.332	0.865	**0.514**
MDS	0.626	0.349	0.695	**0.34**	0.796	0.272	**0.845**	0.295	0.638	0.397
PPMI	**0.808**	0.361	0.732	0.301	0.76	0.242	0.83	0.283	**0.894**	0.312
Topic	0.771	**0.365**	**0.863**	0.3	**0.85**	0.219	0.828	**0.348**	0.819	0.349

Movies Ratings	D1 ACC	D1 F1	D2 ACC	D2 F1	D3 ACC	D3 F1	DN ACC	DN F1	SVM ACC	SVM F1
PCA	**0.65**	0.463	0.681	**0.475**	0.684	**0.486**	0.744	0.408	0.771	0.58
AWV	0.601	0.423	0.618	0.433	0.596	0.448	0.736	0.372	0.73	0.532
MDS	0.592	0.437	0.635	0.449	0.631	0.452	**0.752**	**0.412**	0.773	**0.589**
PPMI	0.583	0.47	0.635	0.453	0.605	0.453	0.73	0.384	**0.825**	0.536
Topic	0.575	**0.473**	**0.789**	0.243	**0.789**	0.38	0.739	0.375	0.704	0.501

3.4.4 Word Directions

Although Linear SVM's perform the best on these representations without the need for interpretability, other results will be for low-depth Decision Trees in-order to easily distinguish the degree to which key semantics correspond to dimensions in the representations.

The main takeaway from this section is that in most cases performance greatly increases compared to the original representations used directly as input to the model (For the exact differences, see Appendix 6.1).

Interestingly, there was also more variance in the difference between space-type sizes, making it an important hyper-parameter for the single directions. The best space type also varied across domains. Loosely, it is possible to attribute the performance increase for a space-type to either modelling the rankings for the same directions better, or containing unique terms that were particularly relevant to the classes. However, when looking at the qualitative results, generally the words common to all space-types are the most salient 3.4. We can see if this is the case by looking at the Decision Trees for the same task that had the most difference between the space-types and space-sizes. If a Decision Tree contains mostly similar words, but the performance is greater, we can attribute it to a better quality ranking in the space. If the Decision Tree contains different words, especially as the first node, then we know that it was because the words that were modelled well were different between them.

We see that generally, the best space type is the same across a variety of tasks in the same domain, AWV is the best for the place-types but MDS is best for the movies (despite a marginal difference in the ratings). This could mean that performance on one natural task will generalize well to the others, so the space-type/size of the space that we identify contains the key semantics for that domain rather than a particular task.

NDCG was selected as the best score-type for Sentiment, Newsgroups, Reuters, Movies Genres, Movies Keywords in depth-3 Decision Trees. Place-types foursquare used F1-score, but the classes are very unbalanced and there are few documents.

Newsgroups	D1				D2				D3			
	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec
PCA 200	0.955	0.348	0.521	0.261	0.959	0.424	0.678	0.309	0.96	0.454	0.674	0.343
PCA 100	0.957	0.382	0.491	0.313	0.961	0.474	0.679	0.364	**0.963**	0.512	0.694	0.406
PCA 50	0.957	0.373	0.417	0.337	**0.963**	0.478	0.621	0.388	0.963	0.506	0.7	0.396
AWV 200	0.832	0.35	0.226	0.777	0.957	0.383	0.517	0.305	0.958	0.445	0.598	0.354
AWV 100	0.83	0.343	0.219	0.785	0.823	0.36	0.233	**0.792**	0.956	0.387	0.563	0.295
AWV 50	0.807	0.341	0.215	0.816	0.833	0.361	0.236	0.762	0.954	0.392	0.511	0.318
MDS 200	**0.959**	**0.418**	**0.543**	0.339	0.962	0.465	0.669	0.357	0.962	0.493	**0.707**	0.379
MDS 100	0.857	0.365	0.244	0.725	0.959	0.428	0.624	0.326	0.96	0.453	0.644	0.349
MDS 50	0.821	0.324	0.206	0.762	0.842	0.386	0.258	0.77	0.957	0.398	0.596	0.299
D2V 200	0.831	0.343	0.22	0.784	0.96	0.47	**0.683**	0.358	0.962	0.494	0.69	0.385
D2V 100	0.844	0.374	0.243	0.803	0.961	**0.49**	0.642	0.396	0.962	0.517	0.67	0.421
D2V 50	0.845	0.388	0.252	**0.844**	0.962	0.488	0.639	0.395	0.963	**0.537**	0.673	**0.446**

Reuters	D1		D2		D3		Sentiment	D1		D2		D3	
	ACC	F1	ACC	F1	ACC	F1		ACC	F1	ACC	F1	ACC	F1
PCA	0.976	0.658	0.979	0.679	0.977	0.467	PCA	0.739	0.759	**0.797**	**0.814**	0.802	0.805
AWV	0.975	0.598	0.979	0.656	0.98	0.66	AWV	0.7	0.699	0.711	0.736	0.723	0.735
MDS	0.975	**0.678**	**0.98**	**0.706**	**0.982**	**0.72**	D2V	**0.776**	**0.784**	0.782	0.801	**0.822**	**0.821**
D2V	**0.977**	0.583	0.979	0.664	0.98	0.632							
Placetypes	D1		D2		D3		Movies	D1		D2		D3	
OpenCYC	ACC	F1	ACC	F1	ACC	F1	Genres	ACC	F1	ACC	F1	ACC	F1
PCA	0.632	0.371	0.704	0.381	0.735	0.365	PCA	0.824	0.412	0.82	0.441	0.913	0.463
AWV	**0.66**	**0.404**	**0.734**	**0.428**	**0.755**	**0.39**	AWV	0.81	0.421	0.837	0.436	0.912	0.457
MDS	0.658	0.374	0.711	0.385	0.746	0.35	MDS	**0.849**	**0.446**	**0.839**	**0.463**	**0.918**	**0.495**
Foursquare	ACC	F1	ACC	F1	ACC	F1	Keywords	ACC	F1	ACC	F1	ACC	F1
PCA	0.785	0.477	**0.907**	0.474	0.869	**0.531**	PCA	0.737	0.225	0.727	0.227	**0.709**	0.22
AWV	**0.918**	**0.541**	0.881	**0.498**	0.889	0.466	AWV	0.656	0.201	0.672	0.203	0.652	0.2
MDS	0.82	0.416	0.879	0.482	**0.897**	0.485	MDS	**0.745**	**0.23**	**0.74**	**0.233**	0.708	**0.224**
Geonames	ACC	F1	ACC	F1	ACC	F1	Ratings	ACC	F1	ACC	F1	ACC	F1
PCA	0.665	0.348	0.754	0.342	0.743	0.306	PCA	**0.647**	**0.466**	**0.721**	**0.499**	0.681	0.492
AWV	**0.711**	**0.444**	**0.795**	**0.533**	**0.802**	**0.473**	AWV	0.646	0.463	0.692	0.474	0.677	0.483
MDS	0.591	0.289	0.772	0.333	0.764	0.352	MDS	0.62	0.463	0.692	0.489	**0.686**	**0.498**

Table 3.10: all dirs

3.4.5 Clustered Directions

?? These results were obtained by taking the single directions that performed the best in the previous results and clustering them with a variety of hyper-parameters for the clusters. K-means mostly outperforms Derrac. It does not in the case of Keywords, where it performs better for every Decision Tree. Although the differences in absolute values are quite small in this case, it is still significant as it is quite difficult to achieve high performance on this task, making these relative changes important. This case can give us insight into how disentanglement affects performance on different classes and domains - and how our unsupervised method selects the best parameters.

When looking into the how the individual classes fared, the 100-size Derrac clusters performed better at the keywords "shot-in-the-chest" and "machine-gun" and sacrificed performance in the "sequel" class. In Derrac, there was the following cluster ("soldiers combat fighting military battle ... weapons rambo gunfights spaghetti guns ...") while in the best performing k-means 200-size clusters these words were split into two separate clusters, one for guns ("gun explosions shoot shooting weapons ... rambo") and one for military ("war soldiers combat military ... platoon infantry"). It's possible that as the Derrac method combined these together into their own cluster they were able to better capture the classes for "shot-in-the-chest" and "machine-guns" because these things occurred in war films where people were shot or shooting. So in this case, the parameters chosen for Derrac supported the classification of the documents into keywords because they better captured particular class concepts through a lesser degree of disentanglement. This idea is supported when looking at the depth-three tree for this class, which uses this cluster as its first node as well as a node in the depth-two layer. This is an instance where having a heavily populated cluster average their direction performs better than strongly disentangling the concepts.

Meanwhile, this same lack of disentanglement caused it to lose performance in the "sequel" class. In K-means, the cluster was found for ("franchise sequels sequel installments") while in Derrac the cluster was ("franchise sequels sequel instalments entry returns"). This cluster was also chosen in Derrac as the first node of its Decision Tree, but this caused it to perform worse than k-means. This is likely because although the words "entry" and "returns" were most similar to this cluster, they disrupted the direction too much. Indeed, when looking at the k-means clusters, the "returns" direction is clustered with "events situation conclusion spoiler ... protagonists exscapes break scenario ...", seemingly referring to a character or thing "returning" in a conclusive part of the movie, and the word "entry" is clustered with the words "effective genuine ... hits build surprisingly ... succeeds essentially finale entry ..." seemingly relating to a more sentiment related cluster about how a movie performed. So in this case k-means being able to find more disentangled clusters than Derrac gave it a performance advantage.

This could be due to the best-performing Derrac clusters being 100-size (meaning the clusters would contain more terms) and the k-means being 200-size. However, in the 100-size K-means clusters, "gun" and "explosions" ended up being in a cluster with ("western outlaw heist shootout west"), making it a more western oriented cluster, and the idea of a war was even more disentangled with a single cluster corresponding to ("war soldiers military solider army sergeant sgt platoon infantry"). In conclusion, Derrac for the Keywords task captured certain concepts better than k-means, in particular by clustering together the idea of "war" and "guns" to achieve high performance on the keywords "shot-in-the-chest" and "machine-guns". K-means favoured a more disentangled approach to these ideas, which meant that although it captured the idea of "war" well, it was not able to capture the classes inbetween the idea of "war" and "guns".

In conclusion, the clustering method that performs the best for a task in this unsupervised context is the one that creates clusters that correspond closely with the task's classes, through clustering together words which average into a particular concept, or disentangling words into concepts so that they more precisely model it.

Newsgroups	D1				D2				D3			
	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec
K-means 200	**0.852**	**0.394**	**0.261**	0.795	**0.958**	**0.433**	**0.58**	0.345	**0.963**	**0.513**	**0.704**	0.403
K-means 100	0.842	0.388	0.257	0.791	0.958	0.366	0.516	0.284	0.962	0.5	0.635	**0.412**
K-means 50	0.834	0.381	0.248	**0.819**	0.815	0.336	0.212	**0.81**	0.961	0.485	0.612	0.402
Derrac 200	0.803	0.313	0.202	0.693	0.797	0.306	0.191	0.781	0.958	0.409	0.605	0.309
Derrac 100	0.792	0.305	0.197	0.667	0.791	0.287	0.179	0.721	0.957	0.374	0.56	0.281
Derrac 50	0.769	0.26	0.162	0.661	0.768	0.237	0.143	0.693	0.955	0.315	0.47	0.237

Table 3.11: All clustering size results for the newsgroups

Reuters	D1		D2		D3		Sentiment	D1		D2		D3	
	ACC	F1	ACC	F1	ACC	F1		ACC	F1	ACC	F1	ACC	F1
K-means	**0.875**	**0.338**	**0.975**	**0.54**	0.973	**0.58**	K-means	0.623	0.674	**0.837**	**0.844**	0.658	0.707
Derrac	0.797	0.291	0.973	0.402	**0.974**	0.485	Derrac	**0.712**	**0.735**	0.802	0.82	**0.803**	**0.813**
Placetypes	D1		D2		D3		Movies	D1		D2		D3	
OpenCYC	ACC	F1	ACC	F1	ACC	F1	Genres	ACC	F1	ACC	F1	ACC	F1
K-means	**0.641**	**0.413**	**0.735**	**0.405**	0.75	**0.43**	K-means	**0.813**	**0.431**	**0.913**	**0.513**	**0.913**	**0.506**
Derrac	0.605	0.39	0.672	0.392	**0.755**	0.391	Derrac	0.759	0.341	0.789	0.431	0.911	0.432
Foursquare	ACC	F1	ACC	F1	ACC	F1	Keywords	ACC	F1	ACC	F1	ACC	F1
K-means	**0.913**	**0.462**	**0.911**	**0.5**	**0.891**	**0.511**	K-means	0.667	0.208	0.648	0.202	0.678	0.213
Derrac	0.768	0.392	0.835	0.445	0.805	0.425	Derrac	**0.726**	**0.215**	**0.745**	**0.22**	**0.707**	**0.219**
Geonames	ACC	F1	ACC	F1	ACC	F1	Ratings	ACC	F1	ACC	F1	ACC	F1
K-means	**0.772**	0.43	**0.774**	0.407	**0.819**	**0.472**	K-means	**0.671**	**0.504**	0.638	**0.507**	**0.686**	**0.513**
Derrac	0.678	**0.449**	0.74	**0.411**	0.807	0.415	Derrac	0.651	0.445	**0.669**	0.463	0.627	0.479

Table 3.12: The best clustering results for each domain and task

3.4.6 Conclusion

In conclusion, we introduce a methodology to go from a Vector Space Model of Semantics and an associated bag-of-words to an interpretable representation and interpretable classifiers. We define an interpretable representation in this work as having two properties: disentanglement and labels, and an interpretable classifier as a simple linear classifier that has components corresponding to the interpretable representation that has these properties, e.g. nodes in a decision tree. In general, we give a simple methodology that can be used to achieve interpretable features and classifiers as an alternative to methods like Topic Models, and give insight into the parameters required and qualitative results that can be obtained. We extensively test the qualitative and quantitative results, finding that the highest-performing quantitative results also make good intuitive qualitative sense. We find that our method greatly outperforms the original representations on low-depth Decision Trees, giving good evidence that we have disentangled the representation. Additionally, we find that we are also competitive with standard interpretable representation baselines in most cases. We introduce variations to the original work that produced these kind of interpretable representations, in particular finding that scoring directions using NDCG performed better than Kappa in most cases, and that we could achieve much stronger results than the original clustering method using K-means. Further, we experimented using a variety of space-types and domains, verifying that the methodology can be applied more generally than shown in [11]. The main experiments that would be interesting to expand on for this chapter would be more state-of-the-art representations, specific investigations of how those representations are able to achieve such strong results, and interpretability experiments to see how our cluster labels fare in real-world situations.

Chapter 4

Fine-tuning Vector Spaces to Improve Their Directions

4.1 Introduction

In this Chapter, we focus on improving the representation of feature directions in domain-specific semantic spaces. Feature directions in Chapter 3 emerged from vector space representations that have been learned with a similarity-centred objective, i.e. the main consideration when learning these representations is that similar objects should be represented as similar vectors. An important observation is that such spaces may not actually be optimal for modelling feature directions. For example, [40] found that PCA dimensions did not typically make sense, and FRAGE [16] achieved state-of-the-art results by removing the bias that a representation has for frequent words. If these biases exist, and the structure is influenced by the similarity such that it does not prioritize a directional structure, then it is unlikely to achieve good results with the method from Chapter 3. To illustrate why this can be the case, Figure 4.1 shows a toy example in which basic geometric shapes are embedded in a two-dimensional space. Within this space, we can identify directions which encode how light an object is and how closely its shape resembles a square. While most of the shapes embedded in this space are grey-scale circles and squares, one of the shapes embedded in this space is a red triangle, which is a clear outlier. If this space is learned with a similarity-centred objective, the representation of the triangle will be far from all the other shapes. However, this means that outliers like this will often take up extreme positions in the rankings induced by the feature directions, and may thus lead us to incorrectly assume that they have certain features. In this example, the triangle would incorrectly be considered as the shape which most exhibits the features “light” and “square”. In contrast,

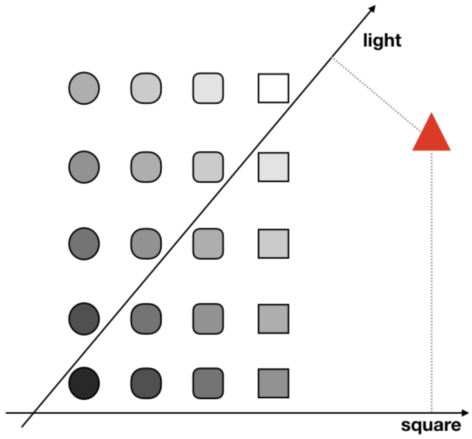

Figure 4.1: Toy example showing the effect of outliers in a two-dimensional embedding of geometric shapes..

if we had learned the representation with the knowledge that it should model these two features rather than similarity, this triangle would have ended up closer to the bottom-left corner.

Unfortunately, we usually have no *a priori* knowledge of which are the most salient features. In this paper, we therefore suggest the following fully unsupervised strategy. First, we learn a semantic space from bag-of-words representations of the considered objects, using a standard similarity-centric method. Using the method from [10], we subsequently determine the most salient features in the considered domain, and their corresponding directions. Finally, we fine-tune the semantic space and the associated feature directions, modelling the considered features in a more faithful way. This last step is the main contribution of this Chapter.

4.2 Related Work

Fine-tuning embeddings. Several authors have looked at approaches for adapting word embeddings. One possible strategy is to change how the embedding is learned in the first place. For example, some approaches have been proposed to learn word embeddings that are better suited at capturing sentiment [49], or to learn embeddings that are optimized for relation extraction [18]. Other approaches, however, start with a pre-trained embedding, which is then modified in a particular way. For example, in [14] a method is proposed to bring the vectors of semantically related words, as specified in a given lexicon, closer together. Similarly [60] propose a

20 Newsgroups: Accuracy Scored	Movie Reviews: NDCG Scored	Place-types: Kappa Scored
{sins, sinful, jesus, moses}	{environmentalist, wildlife, ecological}	{smile, kid, young, female}
{hitters, catcher, pitching, batting}	{prophets, bibles, scriptures}	{rust, rusty, broken, mill}
{ink, printers, printer, matrix}	{assassinating, assasins, assasin}	{eerie, spooky, haunted, ghosts}
{jupiter, telescope, spacecraft, satellites}	{reanimated, undead, zombified}	{religious, christian, chapel, carved}
{firearm, concealed, handgun, handguns}	{ufos, ufo, extraterrestrial, extraterrestrials}	{fur, tongue, teeth, ears}
{escaped, terror, wounded, fled}	{swordsman, feudal, swordfight, swordplay}	{weeds, shed, dirt, gravel}
{cellular, phones, phone}	{scuba, divers, undersea}	{stonework, archway, brickwork}
{brake, steering, tires, brakes}	{regiment, armys, soliders, infantry}	{rails, rail, tracks, railroad}
{riders, rider, ride, riding}	{toons, animations, animating, animators}	{dirty, trash, grunge, graffiti}
{formats, jpeg, gif, tiff}	{fundamentalists, doctrine, extremists}	{tranquility, majestic, picturesque}
{physicians, treatments physician}	{semitic, semitism, judaism, auschwitz}	{monument, site, arch, cemetery}
{bacteria, toxic, biology, tissue}	{shipwrecked, ashore, shipwreck}	{journey, traveling, travelling}
{planets, solar, mars, planetary}	{planetary, earths, asteroid, spaceships}	{mother, mom, children, child}
{symptoms, syndrome, diagnosis}	{atheism, theological, atheists, agnostic}	{frost, snowy, icy, freezing}
{universities, nonprofit, institution}	{astronaut, nasa, spaceship, astronauts}	{colourful, vivid, artistic, vibrant}

Table 4.1: The first clustered words of features for three different domains and three different scoring types. .

method for refining word vectors to improve how well they model sentiment. In [28] a method is discussed to adapt word embeddings based on a given supervised classification task.

4.3 Identifying Feature Directions

4.4 Fine-Tuning Feature Directions

To illustrate that the method from Section 3.2 can produce sub-optimal directions, the second column of Table 4.2 shows the top-ranked objects for some feature directions in the semantic space of place-types. For the feature represented by the cluster $\{steep, climb, slope\}$, the top

Feature direction	Highest ranking objects	Highest fine-tuned ranking objects
{steep, climb, slope}	mountain, landscape, national park	ski slope, steep slope, slope
{illuminated, illumination, skyscraper}	building, city, skyscraper	tall building, office building, large building
{play, kid, kids}	school, field, fence	college classroom, classroom, school
{spooky, creepy, scary}	hallway, fence, building	hospital room, hospital ward, patient room
{amazing, dream, awesome}	fence, building, beach	hotel pool, resort, beach resort
{pavement, streetlight, streets}	sidewalk, fence, building	overpass road, overpass, road junction
{dead, hole, death}	fence, steps, park	grave, cemetery, graveyard
{spire, belltower, towers}	building, arch, house	bell tower, arch, religious site
{stones, moss, worldheritage}	landscape, fence, steps	ancient site, ancient wall, tomb
{mosaic, tile, bronze}	building, city, steps	cathedral, church, religious site

Table 4.2: Comparing the highest ranking place-type objects in the original and fine-tuned space. .

ranked object *mountain* is clearly relevant. However, the next two objects — *landscape* and *national park* — are not directly related to this feature. Intuitively, they are ranked highly because of their similarity to *mountain* in the vector space. Similarly, for the second feature, *building* is ranked highly because of its similarity to *skyscraper*, despite intuitively not having this feature. Finally, *fence* received a high rank for several features, mostly because it is an outlier in the space.

To improve the directions and address these problems, we propose a method for fine-tuning the semantic space representations and corresponding feature directions. The main idea is to use the BoW representations of the objects as a kind of weak supervision signal: if an object should be ranked highly for a given feature, we would expect the words describing that feature to appear frequently in its description. In particular, for each feature f we determine a total ordering $\preccurlyeq_f$ such that $o \preccurlyeq_f o'$ iff the feature f is more prominent in the BoW represention of object o' than in the BoW representation of o. We will refer to $\preccurlyeq_f$ as the *target ranking* for feature f. If the feature directions are in perfect agreement with this target ranking, it would be the case that $o \preccurlyeq o'$ iff $v_C \cdot o \leq v_C \cdot o'$. Since this will typically not be the case, we subsequently determine *target values* for the dot products $v_C \cdot o$. These target values represent the minimal way in which the dot products need to be changed to ensure that they respect the target ranking. Finally, we use a simple feedforward neural network to adapt the semantic space representations o and feature directions v_C to make the dot products $v_C \cdot o$ as close as possible to these target values.

4.4.1 Generating Target Rankings

Let $C_1, ..., C_K$ be the clusters that were found using the method from Section 4.3. Each cluster C_i typically corresponds to a set of semantically related words $\{w_1, ..., w_n\}$, which describe some salient feature from the considered domain. From the BoW representations of the objects, we can now define a ranking that reflects how strongly each object is related to the words from this cluster. To this end, we represent each object as a bag of clusters (BoC) and then compute PPMI scores over this representation. In particular, for a cluster $C = \{w_1, ..., w_m\}$, we define $n(C, o) = \sum_{i=1}^{m} n(w_i, o)$. In other words, $n(C, o)$ is the total number of occurrences of words from cluster C in BoW representation of o. We then write *ppmi*(C, o) for the PPMI score corresponding to this BoC representation, which is evaluated in the same way as *ppmi*(C, o), but using the counts $n(C, o)$ rather than $n(w, o)$. The target ranking for cluster C_i is then such that o_1 is ranked higher than o_2 iff *ppmi*$(C_i, o_1) >$ *ppmi*(C_i, o_2). By computing PPMI scores w.r.t. clusters of words, we alleviate problems with sparsity and synonymy, which in turn allows us to better estimate the intensity with which a given feature applies to the object. For instance, an object describing a violent movie might not actually mention the word ‘violent’, but would likely mention at least some of the words from the same cluster (e.g. ‘bloody’ ‘brutal’ ‘violence’ ‘gory’). Similarly, this approach allows us to avoid problems with ambiguous word usage; e.g. if a movie is said to contain ‘violent language’, it will not be identified as violent if other words related to this feature are rarely mentioned.

4.4.2 Generating Target Feature Values

Finding directions in a vector space that induce a set of given target rankings is computationally hard[1]. Therefore, rather than directly using the target rankings from Section 4.4.1 to fine-tune the semantic space, we will generate target values for the dot products $v_{C_j} \cdot o_i$ from these target rankings. One straightforward approach would be to use the PPMI scores *ppmi*(C_j, o_i). However these target values would be very different from the initial dot products, which among

[1]It is complete for the complexity class $\exists\mathbb{R}$, which sits between NP and PSPACE [47].

others means that too much of the similarity structure from the initial vector space would be lost. Instead, we will use isotonic regression to find target values $\tau(C_j, o_i)$ for the dot product $v_{C_j} \cdot o_i$, which respect the ranking induced by the PPMI scores, but otherwise remain as close as possible to the initial dot products.

Let us consider a cluster C_j for which we want to determine the target feature values. Let $o_{\sigma_1}, ..., o_{\sigma_n}$ be an enumeration of the objects such that *ppmi*$(C_j, o_{\sigma_i}) \leq$ *ppmi*$(C_j, o_{\sigma_{i+1}})$ for $i \in \{1, ..., n-1\}$. The corresponding target values $\tau(C_j, o_i)$ are then obtained by solving the following optimization problem:

Minimize: $\sum_i (\tau(C_j, o_i) - v_{C_j} \cdot o_i)^2$

Subject to:

$\tau(C_j, o_{\sigma_1}) \leq \tau(C_j, o_{\sigma_2}) \leq ... \leq \tau(C_j, o_{\sigma_n})$

4.4.3 Fine-Tuning

We now use the target values $\tau(C_j, o_i)$ to fine-tune the initial representations. To this end, we use a simple neural network architecture with one hidden layer. As inputs to the network, we use the initial vectors $o_1, ..., o_n \in \mathbb{R}^k$. These are fed into a layer of dimension l:

$$h_i = f(Wo_i + b)$$

where W is an $l \times k$ matrix, $b \in \mathbb{R}^l$ is a bias term, and f is an activation function. After training the network, the vector h_i will correspond to the new representation of the i*th* object. The vectors h_i are finally fed into an output layer containing one neuron for each cluster:

$$g_i = Dh_i$$

where D is a $K \times l$ matrix. Note that by using a linear activation in the output layer, we can interpret the rows of the matrix D as the K feature directions, with the components of the vector $g_i = (g_i^1, ..., g_i^K)$ being the corresponding dot products. As the loss function for training the network, we use the squared error between the outputs g_i^j and the corresponding target values

20 Newsgroups	F1 D1	F1 D3	F1 DN
FT MDS	**0.50**	**0.47**	**0.44**
MDS	0.44	0.42	0.43
FT PCA	0.40	0.36	0.34
PCA	0.25	0.27	0.36
FT Doc2Vec	0.44	0.42	0.41
Doc2Vec	0.29	0.34	0.44
FT AWV	0.47	0.45	0.40
AWV	0.41	0.38	0.43
FT AWVw	0.41	0.41	0.43
AWVw	0.38	0.40	0.43
LDA	0.40	0.37	0.35

Table 4.3: Results for 20 Newsgroups.

$\tau(C_j, o_i)$, i.e.:

$$\mathcal{L} = \sum_i \sum_j (g_i^j - \tau(C_j, o_i))^2$$

The effect of this fine-tuning step is illustrated in the right-most column of Table 4.2, where we can see that in each case the top ranked objects are now more closely related to the feature, despite being less common, and outliers such as 'fence' no longer appear.

4.5 Evaluation

To evaluate our method, we consider the problem of learning interpretable classifiers. In particular, we learn decision trees which are limited to depth 1 and 3, which use the rankings induced by the feature directions as input. This allows us to simultaneously assess to what extent the method can identify the right features and whether these features are modelled well using the learned directions. Note that depth 1 trees are only a single direction and a cut-off, so to perform well, the method needs to identify a highly relevant feature to the considered category. Depth 3 decision trees are able to model categories that can be characterized using at most three feature directions.

4.5.1 Experimental set-up

Datasets. We evaluate our method on four datasets. First, we used the *movies* and *place-types* datasets from [10], which are available in preprocessed form[2]. The former describes 15000 movies, using a BoW representation that was obtained by combining reviews from several sources. However, 1022 duplicate movies were found in the data, which we removed. The associated classification tasks are to predict the movie genres according to IMDB (23 classes), predicting IMDB plot keywords such as 'suicide', 'beach' or 'crying' (100 classes) and predicting age rating certificates such as 'UK-15' 'UK-18' or 'USA-R' (6 classes). All tasks are evaluated as binary classification tasks. We randomly split the datasets into 2/3 for training and 1/3 for testing. The place-types dataset was obtained by associating each place-type with the bag of tags that have been used to describe places of that type on Flickr. It contains BoW represenations for 1383 different place-types. The classification problems for this dataset involve predicting whether a place-type belongs to a given category in three different taxonomies: Geonames (7 classes), Foursquare (9 classes) and OpenCYC (20 classes). Since many of these categories are very small, for this dataset we have used 5-fold cross validation.

The remaining two datasets are standard datasets for document classification: *20 newsgroups* and the *IMDB sentiment* dataset. For the 20 newsgroups dataset, the standard[3] split was used where 11314 of the 18446 documents are used for training. Headers, footers and quote metadata were removed using scikit-learn[4]. The associated classification problem is to predict which newsgroup a given post was submitted to (20 classes). The IMDB sentiment dataset contains a total of 50000 documents, and it is split into 25000 documents for training and 25000 for testing. For the newsgroups and sentiment datasets, we used stopwords from the NLTK python package [35]. For these datasets, we used all (lowercased) tokens and retained numbers, rather than only using nouns and adjectives. The associated classification problem is to predict the sentiment of the review (positive or negative).

Semantic Spaces. We will consider semantic spaces that have been learned using a number of different methods. First,

We also consider PCA, which As our third method, we consider Doc2vec, which is

[2]`http://www.cs.cf.ac.uk/semanticspaces/`

[3]`http://qwone.com/~jason/20Newsgroups/`

[4]`http://scikit-learn.org/stable/datasets/twenty_newsgroups.html`

Movie Reviews

Genres	D1	D3	DN	**Keywords**	D1	D3	DN	**Ratings**	D1	D3	DN
FT MDS	**0.57**	**0.56**	0.51	FT MDS	**0.33**	**0.33**	0.24	FT MDS	**0.49**	**0.51**	**0.46**
MDS	0.40	0.49	**0.52**	MDS	0.31	0.32	**0.25**	MDS	0.46	0.49	**0.46**
FT AWV	0.42	0.42	0.39	FT AWV	0.25	0.25	0.15	FT AWV	0.47	0.44	0.39
AWV	0.35	0.44	0.43	AWV	0.26	0.21	0.19	AWV	0.44	0.48	0.41
LDA	0.52	0.51	0.45	LDA	0.22	0.19	0.18	LDA	0.48	0.48	0.41

Place-types

Geonames	D1	D3	DN	**Foursquare**	D1	D3	DN	**OpenCYC**	D1	D3	DN
FT MDS	0.32	0.31	0.24	FT MDS	0.41	0.44	0.41	FT MDS	0.35	0.36	0.30
MDS	0.32	0.31	0.21	MDS	0.38	0.42	0.42	MDS	0.35	0.36	0.29
FT AWV	0.31	0.29	0.23	FT AWV	0.39	0.42	0.41	FT AWV	0.37	**0.37**	0.28
AWV	0.28	0.28	0.22	AWV	0.32	0.37	0.31	AWV	0.33	0.35	0.26
LDA	**0.34**	**0.32**	**0.27**	LDA	**0.55**	**0.48**	**0.47**	LDA	**0.40**	0.36	**0.31**

Table 4.4: The results for Movie Reviews and Place-Types on depth-1, depth-3 and unbounded trees. .

Methodology.

We train the network using AdaGrad [12], with default values, and the model was implemented in the Keras library.

4.5.2 Results

Table 4.3 shows the results for the 20 newsgroups dataset, where we use FT to indicate the results with fine-tuning[5]. We can see that the fine-tuning method consistently improves the per-

[5]Since the main purpose of this first experiment was to see whether fine-tuning improved consistently across a broad set of representations, here we considered a slightly reduced pool of parameter values for hyperparameter tuning.

IMDB Sentiment	D1	D3	DN
FT PCA	0.78	0.80	0.79
PCA	0.76	**0.82**	**0.80**
FT AWV	0.72	0.76	0.71
AWV	0.74	0.76	0.71
LDA	**0.79**	0.80	0.79

Table 4.5: Results for IMDB Sentiment.

formance of the depth-1 and depth-3 trees, often in a very substantial way. After fine-tuning, the results are also consistently better than those of LDA. For the unbounded trees (DN), the differences are small and fine-tuning sometimes even makes the results worse. This can be explained by the fact that the fine-tuning method specializes the space towards the selected features, which means that some of the structure of the initial space will be distorted. Unbounded decision trees are far less sensitive to the quality of the directions, and can even perform reasonably on random directions. Interestingly, depth-1 trees achieved the best overall performance, with depth-3 trees and especially unbounded trees overfitting. Since MDS and AWV perform best, we have only considered these two representations (along with LDA) for the remaining datasets, except for the IMDB Sentiment dataset, which is too large for using MDS.

The results for the movies and place-types datasets are shown in Table 4.4. For the MDS representations, the fine-tuning method again consistently improved the results for D1 and D3 trees. For the AWV representations, the fine-tuning method was also effective in most cases, although there are a few exceptions. What is noticeable is that for movie genres, the improvement is substantial, which reflects the fact that genres are a salient property of movies. For example, the decision tree for the genre 'Horror' could use the feature direction for $\{gore, gory, horror, gruesome\}$. Some of the other datasets refer to more specialized properties, and the performance of our method then depends on whether it has identified features that relate to these properties. It can be expected that a supervised variant of this method would perform consistently better in such cases. After fine-tuning, the MDS based representation outperforms LDA on the movies dataset, but not for the place-types. This is a consequence of the fact that some of the place-type categories refer to very particular properties, such as geological phenomena, which may not be particularly dominant among the Flickr tags that were used to generate the spaces. In such cases, using a BoW based representation may be more suitable.

Finally, the results for IMDB Sentiment are shown in Table 4.5. In this case, the fine-tuning method fails to make meaningful improvements, and in some cases actually leads to worse results. This can be explained from the fact that the feature directions which were found for this space are themes and properties, rather than aspects of binary sentiment evaluation. The fine-tuning method aims to improve the representation of these properties, possibly at the expense of other aspects.

4.6 Conclusions

We have introduced a method to identify and model the salient features from a given domain as directions in a semantic space. Our method is based on the observation that there is a trade-off between accurately modelling similarity in a vector space, and faithfully modelling features as directions. In particular, we introduced a post-processing step, modifying the initial semantic space, which allows us to find higher-quality directions. We provided qualitative examples that illustrate the effect of this fine-tuning step, and quantitatively evaluated its performance in a number of different domains, and for different types of semantic space representations. We found that after fine-tuning, the feature directions model the objects in a more meaningful way. This was shown in terms of an improved performance of low-depth decision trees in natural categorization tasks. However, we also found that when the considered categories are too specialized, the fine-tuning method was less effective, and in some cases even led to a slight deterioration of the results. We speculate that performance could be improved for such categories by integrating domain knowledge into the fine-tuning method.

[39] "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 Wikipedia text to enable this decomposition can produce substantial im- provements on semantic-similarity, pos- induction and word-analogy tasks."

Chapter 5

Investigating Neural Networks In Terms Of Directions

5.1 Chapter 5

Neural network models that encode spatial relationships in their hidden layers have achieved state-of-the-art in Text Classification by using transfer learning from a pre-trained Language Model [16]. There have also been neural network models that produce an interpretable representation, for example InfoGan. Most state-of-the-art results rely on Vector Space Models. Ideally the method would be able to achieve strong results for simple interpretable classifiers by transforming an existing representation that performs well at the task.

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.

Chapter 6

Appendix

6.1 Chapter 3

6.1.1 Difference between Representations and Single Directions

Newsgroups	D1				D2				D3			
	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec	ACC	F1	Prec	Rec
PCA 200	0.254	0.097	0.373	-0.55	0.117	0.058	0.433	-0.41	0.004	0.099	0.134	0.078
PCA 100	0.259	0.135	0.345	-0.5	0.126	0.112	0.438	-0.367	0.006	0.157	0.118	**0.149**
PCA 50	0.277	0.133	0.277	-0.492	0.129	0.123	0.387	-0.347	0.006	0.177	0.228	0.143
AWV 200	0.145	0.133	0.1	-0.005	0.199	0.128	0.362	-0.414	0.194	0.185	0.441	-0.397
AWV 100	0.153	0.133	0.098	0.01	0.043	0.084	0.06	**0.109**	0.21	0.137	0.414	-0.474
AWV 50	0.11	0.122	0.088	0.044	0.056	0.088	0.068	0.052	0.21	0.142	0.362	-0.468
MDS 200	**0.378**	**0.234**	**0.439**	-0.498	**0.22**	0.203	0.509	-0.372	0.243	**0.257**	**0.568**	-0.406
MDS 100	0.271	0.178	0.138	-0.108	0.205	0.167	0.465	-0.401	**0.254**	0.217	0.506	-0.459
MDS 50	0.228	0.171	0.119	**0.115**	0.126	0.136	0.108	0.014	0.222	0.155	0.452	-0.476
D2V 200	0.149	0.138	0.101	0.037	0.158	0.202	**0.514**	-0.288	0.192	0.225	0.526	-0.365
D2V 100	0.162	0.166	0.123	0.041	0.169	**0.222**	0.474	-0.266	0.176	0.249	0.505	-0.306
D2V 50	0.162	0.181	0.132	0.08	0.154	0.193	0.452	-0.299	0.181	0.256	0.501	-0.314

Reuters	D1		D2		D3		Sentiment	D1		D2		D3	
	ACC	F1	ACC	F1	ACC	F1		ACC	F1	ACC	F1	ACC	F1
PCA	0.129	0.33	0.062	0.265	-0.002	-0.034	PCA	-0.006	0.053	0.042	0.044	0.024	0.032
AWV	**0.193**	0.345	0.008	0.327	**0.007**	**0.243**	AWV	0.057	0.047	0.068	0.042	0.028	0.018
MDS	0.184	**0.414**	0.08	0.349	0.003	0.231	D2V	**0.134**	**0.12**	**0.122**	**0.094**	**0.12**	**0.121**
D2V	0.159	0.316	**0.112**	**0.366**	0.006	0.188							
Placetypes	D1		D2		D3		Movies	D1		D2		D3	
OpenCYC	ACC	F1	ACC	F1	ACC	F1	Genres	ACC	F1	ACC	F1	ACC	F1
PCA	**0.047**	**0.025**	-0.003	0.038	**0.04**	**0.024**	PCA	0.102	0.111	**0.064**	0.101	**0.196**	**0.142**
AWV	0.036	0.021	**0.083**	**0.052**	0.027	-0.006	AWV	0.132	0.132	0.064	**0.115**	0.156	0.114
MDS	0.034	0.009	0.011	-0.012	0.016	-0.024	MDS	**0.17**	**0.148**	0.049	0.104	0.145	0.141
Foursquare	ACC	F1	ACC	F1	ACC	F1	Keywords	ACC	F1	ACC	F1	ACC	F1
PCA	0.054	0.135	**0.084**	**0.082**	0.008	**0.143**	PCA	0.09	0.04	**0.083**	0.034	0.032	0.022
AWV	**0.151**	**0.14**	0.053	0.02	**0.038**	0.014	AWV	**0.156**	0.041	0.031	0.024	**0.057**	**0.025**
MDS	-0.094	-0.022	0.075	0.055	0.038	0.031	MDS	0.111	**0.051**	0.05	**0.035**	0.033	0.023
Geonames	ACC	F1	ACC	F1	ACC	F1	Ratings	ACC	F1	ACC	F1	ACC	F1
PCA	**0.163**	0.047	0.063	0.037	**0.063**	0.011	PCA	-0.003	0.003	0.04	0.023	-0.003	0.007
AWV	0.054	**0.119**	0.04	**0.21**	-0.039	**0.106**	AWV	**0.045**	**0.041**	**0.074**	**0.042**	**0.08**	0.036
MDS	-0.035	-0.06	**0.078**	-0.007	-0.032	0.08	MDS	0.028	0.026	0.057	0.04	0.055	**0.045**

Table 6.1: The difference between the representations being directly input to the low-depth decision trees and the word directions

6.1.2 Class Names and Positive Occurrences

Newsgroups	Positives	OpenCYC	Positives	Foursquare	Positives	Geonames	Positives	Genres	Positives	Ratings	Positives
alt.atheism	799	aqueduct	67	ArtsAndEntertainment	39	StreamLake	74	Action	2105	USA-G	1974
comp.graphics	973	border	556	CollegeAndUniversity	33	ParksArea	28	Adventure	1451	UK-12-12A	1566
comp.os.ms-windows.misc	985	building	91	Food	82	RoadRailroad	16	Animation	396	UK-15	3957
comp.sys.ibm.pc.hardware	982	dam	389	ProfessionalAndOtherPlaces	47	SpotBuildingFarm	176	Biography	627	UK-18	2009
comp.sys.mac.hardware	963	facility	173	NightlifeSpot	17	MountainHillRock	68	Comedy	4566	UK-PG	1724
comp.windows.x	988	foreground	43	ParksAndOutdoors	44	Undersea	27	Crime	2073	USA-PG-PG13	439
misc.forsale	975	historical_site	297	ShopsAndService	88	ForestHeath	14	Documentary	781	USA-R	5170
rec.autos	990	holy_site	44	TravelAndTransport	35			Drama	7269		
rec.motorcycles	996	landmark	96	Residence	6			Family	873		
rec.sport.baseball	994	medical_facility	28					Fantasy	928		
rec.sport.hockey	999	medical_school	49					Film-Noir	170		
sci.crypt	991	military_place	30					History	502		
sci.electronics	984	monsoon_forest	53					Horror	1963		
sci.med	990	national_monument	145					Music	1051		
sci.space	987	outdoor_location	103					Musical	529		
soc.religion.christian	997	rock_formation	184					Mystery	1128		
talk.politics.guns	910	room	60					Romance	2965		
talk.politics.mideast	940							Sci-Fi	1266		
talk.politics.misc	775							Short	560		
talk.religion.misc	628							Sport	385		
								Thriller	3293		
								War	671		
								Western	454		

Keywords (1)	Positives	Keywords (2)		Keywords (3)		Keywords (4)		Keywords (5)		Reuters	Positives
adultery	853	dancing	1655	funeral	802	money	887	shot-to-death	976	trade	466
bar	1334	death	2596	gore	820	mother-daughter-relationship	1477	singer	1278	grain	580
bare-breasts	1360	doctor	1193	gun	1445	mother-son-relationship	1908	singing	1372	nat-gas	105
bare-chested-male	1360	dog	1605	gunfight	776	murder	3496	song	986	crude	568
based-on-novel	2390	drink	1080	helicopter	864	new-york-city	1464	suicide	1092	sugar	162
beach	881	drinking	1246	hero	789	nudity	1887	surprise-ending	1202	corn	237
beating	1011	drunkenness	1291	horse	825	one-word-title	1357	tears	892	veg-oil	124
betrayal	848	escape	789	hospital	1434	party	1131	telephone-call	1187	ship	280
blood	2384	explosion	1283	hotel	902	photograph	1304	title-spoken-by-character	1725	coffee	139
boy	824	face-slap	907	husband-wife-relationship	2392	pistol	1378	topless-female-nudity	1079	wheat	283
boyfriend-girlfriend-relationship	1093	falling-from-height	875	independent-film	3431	police	1801	train	1069	gold	120
brother-brother-relationship	884	family-relationships	1787	infidelity	862	policeman	792	underwear	860	acq	2363
brother-sister-relationship	1025	father-daughter-relationship	1758	jealousy	928	pregnancy	821	violence	2231	interest	457
character-name-in-title	2146	father-son-relationship	2201	kidnapping	863	punched-in-the-face	870	voice-over-narration	1058	money-fx	676
chase	1351	female-nudity	2328	kiss	1759	rain	1053	watching-tv	887	soybean	111
church	897	fight	1356	knife	1097	restaurant	1202	wedding	800	oilseed	171
cigarette-smoking	1858	fire	1027	love	2164	revenge	1336			earn	3951
corpse	1008	fistfight	977	machine-gun	878	sequel	801			bop	104
crying	1149	flashback	1937	male-nudity	1122	sex	2126			gnp	136
cult-film	1636	friend	1193	marriage	1407	shootout	1174			dlr	162
dancer	1020	friendship	1903	martial-arts	824	shot-in-the-chest	892			money-supply	168

Table 6.2: Positive Instance Counts for each Class

GNU Free Documentation License

Version 1.2, November 2002

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] David Arthur and Sergei Vassilvitskii. k-means ++ : The Advantages of Careful Seeding. 8:1–11.

[2] 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.

[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, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[5] Andrew M. Dai, Christopher Olah, and Quoc V. Le. Document embedding with paragraph vectors. *CoRR*, abs/1507.07998, 2015.

[6] Andrew M Dai, Christopher Olah, and Quoc V. Le. Document Embedding with Paragraph Vectors. pages 1–8, 2015.

[7] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proc. ACL*, pages 795–804, 2015.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[9] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In *IEEE International Conference on Computer Vision*, pages 1241–1250, 2017.

[10] J. Derrac and S. Schockaert. Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artificial Intelligence*, pages 74–105, 2015.

[11] 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.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[13] Sergey Edunov, Myle Ott, Michael Auli, David Grangier, Menlo Park, Google Brain, and Mountain View. Understanding Back-Translation at Scale. 2018.

[14] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, 2015.

[15] Manaal Faruqui and Chris Dyer. Non-distributional Word Vector Representations. *Acl-2015*, pages 464–469, 2015.

[16] Chengyue Gong. FRAGE : Frequency-Agnostic Word Representation. 1(Nips):1–15, 2018.

[17] Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian PadÃ^3$.Distributionalvectorsencodereferentialattributes.InProceedingsofthe2015Conferenceo$

[18] Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. Task-oriented learning of word embeddings for semantic relation classification. *CoRR*, abs/1503.00095, 2015.

[19] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, 2016.

[20] 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.

[21] H.˜Zou, T.˜Hastie, R.˜Tibshirani, Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[22] Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. MEmbER : Max-Margin Based Embeddings for Entity Retrieval.

[23] Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. Member: Max-margin based embeddings for entity retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 783–792, 2017.

[24] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[25] Joo-kyung Kim. Deriving adjectival scales from continuous space word representations. (October):1625–1630, 2013.

[26] Adriana Kovashka, Devi Parikh, and Kristen Grauman. WhittleSearch : Image Search with Relative Attribute Feedback.

[27] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2973–2980, 2012.

[28] Igor Labutov and Hod Lipson. Re-embedding words. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 489–493, 2013.

[29] Jey Han Lau and Timothy Baldwin. Practical Insights into Document Embedding Generation. 2014.

[30] Quoc Le, Tomas Mikolov, and Tmikolov Google Com. Distributed Representations of Sentences and Documents. 32, 2014.

[31] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1188–1196, 2014.

[32] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 59–66, 2016.

[33] Yang Liu and Mirella Lapata. Learning Structured Text Representations. 2017.

[34] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *Proc. AAAI*, pages 2418–2424, 2015.

[35] Edward Loper and Steven Bird. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, 2002.

[36] Hongyin Luo, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Online Learning of Interpretable Word Embeddings. (September):1687–1692, 2015.

[37] Matej Martinc, Jan Kralj, and Senja Pollak. tax2vec : Constructing Interpretable Features from Taxonomies for Short Text Classification.

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[39] 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.

[40] Brian Murphy, Partha Pratim, and Talukdar Tom. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding.

[41] 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.

[42] Alexander Panchenko. Best of Both Worlds: Making Word Sense Embeddings Interpretable. *the 10th edition of the Language Resources and Evaluation Conference (LREC 2016)*, pages 2649–2655, 2016.

[43] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

[44] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[45] Sascha Rothe and Language Processing. Word Embedding Calculus in Meaningful Ultradense Subspaces. pages 512–517, 2016.

[46] 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.

[47] Steven Schockaert and Jae Hee Lee. Qualitative reasoning about directions in semantic spaces. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 3207–3213. AAAI Press, 2015.

[48] D Sculley. Web-Scale K-Means Clustering. pages 4–5, 2010.

[49] Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28:496–509, 2016.

[50] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 1385–1392, Cambridge, MA, USA, 2004. MIT Press.

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

[52] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 165–174, 2016.

[53] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. Structural regularities in text-based entity vector spaces. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 3–10, 2017.

[54] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 225–232, 2016.

[55] Paolo Viappiani. Preference-based Search using Example-Critiquing with Suggestions. 27:465–503, 2006.

[56] Paolo Viappiani, Boi Faltings, and Pearl Pu. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research*, 27:465–503, 2006.

[57] Jesse Vig, Shilad Sen, and John Riedl. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems*, 2(3):13:1–13:44, 2012.

[58] Jesse Vig, Shilad Sen, and John Riedl. The Tag Genome : Encoding Community Knowledge to Support Novel Interaction The Tag Genome : Encoding Community Knowledge to Support. (November), 2014.

[59] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[60] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 534–539, 2017.

[61] Youwei Zhang and Laurent El Ghaoui. Large-Scale Sparse Principal Component Analysis with Application to Text Data. pages 1–8, 2012.