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

Chapter 1

Introduction

1.1 Motivation

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.

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' [12] also has many positive effects for its users, like lower response times [11, 8], better question answering and confidence for logical problem questions [8] and higher satisfaction [11].
- 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 [5]: 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

2.1.1 Bag-of-words

- Frequency
- Tf-idf
- PPMI

2.2 Text classification

2.2.1 Decision Trees

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

2.2.2 Support Vector Machines

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

2.2.3 Neural Networks

- Difference between SVM and Nnet

2.2.4 Word Representations

- Word-vectors

2.2.5 Document Representations

Conceptual spaces

- PCA
- MDS

2.3 Interpretable Representations

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

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

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

Topic models such as Latent Dirichlet Allocation (LDA) represent documents as multinomial distributions over latent topics, where each of these topics corresponds to a multinomial distribution over words [1]. These topics tend to correspond to semantically meaningful concepts, hence topic models tend to be rather interpretable [4]. To characterize the semantic concepts associated with the learned topics, topics are typically labelled with the most probable words according to the corresponding distribution.

Chapter 3

Converting Vector Spaces into Interpretable Representations

3.1 Introduction

Distributional representations like vector spaces have the ability to represent semantic information spatially. Vector spaces built from a domain-specific corpus of text that represent this kind of information, also known as 'semantic spaces', are used, for instance, to represent items in recommender systems [**?**, **?**, **?**], to represent entities in semantic search engines [**?**, **?**], or to represent examples in classification tasks [**?**]. One way to understand how a semantic space represents information is as a conceptual space [7]. In this space, we can understand domain entities, e.g. movies in a domain of IMDB movie reviews, to be represented as points, and domain properties to be in regions around these points. In figure 3.1, we show an example conceptual space for movies.

The success of these semantic spaces has lead many to investigate how the similarity-based structure can be converted into formal relationships. For example, in word-vectors (see Section **??** for more) [13] found that "equivalent relations tended to correspond to parralel vector differences" [10], while [10] discovered that by decomposing representations into orthogonal semantic and syntactic subspaces they were able to produce substantial improvements on various tasks.

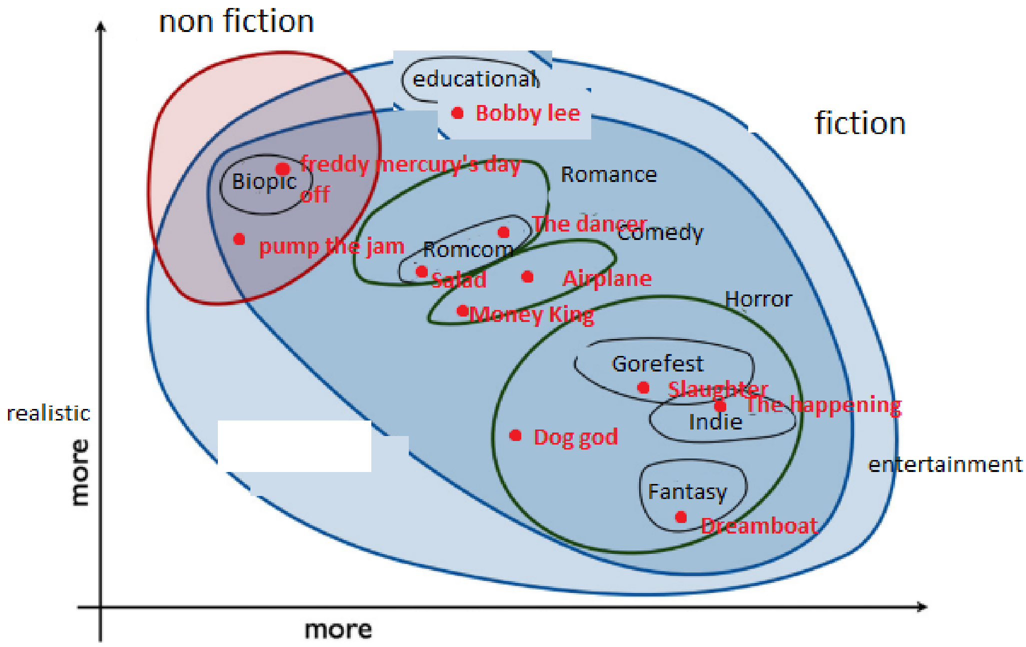

Figure 3.1: A conceptual space of movies, where regions correspond to properties and entities are points..

The semantic relation that we focus in on this paper are directions that correspond to salient features from the considered domain. A direction is the orthogonal direction to a hyper plane that separates a term in a vector space. As the hyper plane separates entities, this means that the entities furthest along the hyper plane, at the end classified positively, are the entities we are most sure have the term we found the hyper plane for. To see an example of this, see **??** With this understanding, it becomes possible to induce a ranking of entities on the properties by finding the dot product of the entity points on the direction vector.

These kind-of directions have been used in many different ways for different domains, For instance, [?] found that features of countries, such as their GDP, fertility rate or even level of CO_2 emissions, can be predicted from word embeddings using a linear regression model. Similarly, in [?] directions in word embeddings were found that correspond to adjectival scales (e.g. bad $<$ okay $<$ good $<$ excellent) while [?] found

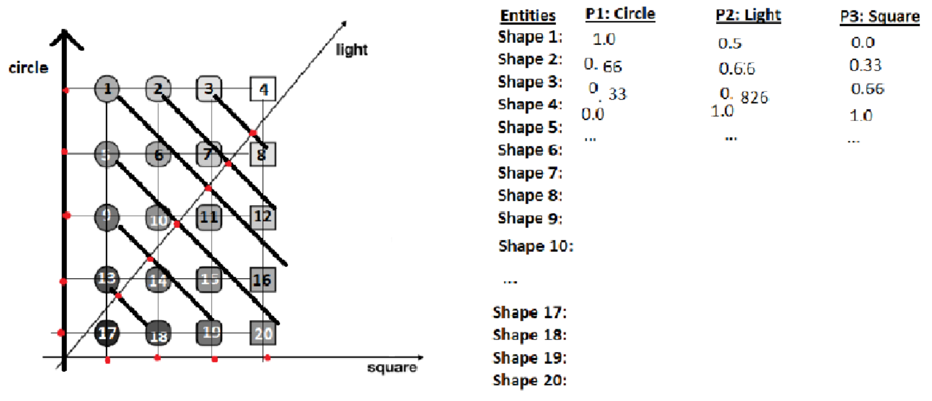

Figure 3.2: This figure shows a 2d toy space where entities are shapes and directions are properties. We demonstrate on the right the method to induce a ranking from the directions, in particular by using the dot-product of the entity point on the directions vector. In the same way for a more complex space, we can understand each entity point to be ranked on thousands of property directions, and the space to be much higher dimensionality..

directions indicating lexical features such as the frequency of occurrence and polarity of words.

By finding the dot product between entity points in the space and direction vectors, it is possible to induce a ranking of entities on those directions. In this chapter, we more deeply investigate the potential of direction vectors to rank entities on properties to form an interpretable representation.

In this thesis, we refer to these direction vectors as directions to convey the ordinal meaning, and directions as 'properties' if they are sufficiently salient in the space, e.g. In a domain of IMDB movie reviews where movies are entities, a direction on the word "The" would not be a property, but a direction on the word "Horror" would be.

Such properties 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 [?]. For instance, [?] propose a movie recommendation system in which the user can specify that they want to see suggestions for movies that are "similar to this one, but scarier". If the property of being scary is adequately modelled as a direction in a semantic space of movies, such critiques can be addressed in a straightforward way. Similarly, in [?] a system was developed that can find "shoes like these but shinier", based on a semantic space representation that was derived from visual features. Semantic search systems can use such directions to interpret queries involving gradual and possibly ill-defined features, such as "*popular* holiday destinations in Europe" [?]. While features such as popularity are typically not encoded in traditional knowledge bases, they can often be represented as semantic space directions.

We demonstrate the effect of different filtering methods to find properties, the ability of different clustering methods to label properties, as well as the number and types of directions, for use in a low-depth interpretable linear classifier; a Decision Tree. In Figure 3.3, we demonstrate how depth could affect a Decision Tree that uses salient properties. These trees are not only evaluated quantitatively on key domain tasks, we also evaluate how interpretable the resulting rules are. This gives us a comprehensive idea of how we can use these rankings as an interpretable representation. By using a Decision Tree, we can identify salient properties - if we are able to construct a simple but high-scoring classifier for if a movie is a 'Comedy' using only our ranking of entities on the property $p = "Funny", "Hilarious", "Laughing"$ then we know that this property is salient. Although this is an extreme case, for more complex concepts, if we have salient properties that form the building blocks of this concept, then the model can be less complex and more general, two desirable properties for interpretable classifiers.

In a case study by [14], giving the business users the option between a model with higher classification score but more input variables and a lower classification score but less input variables resulted in more buy-in for system designers. By accurately

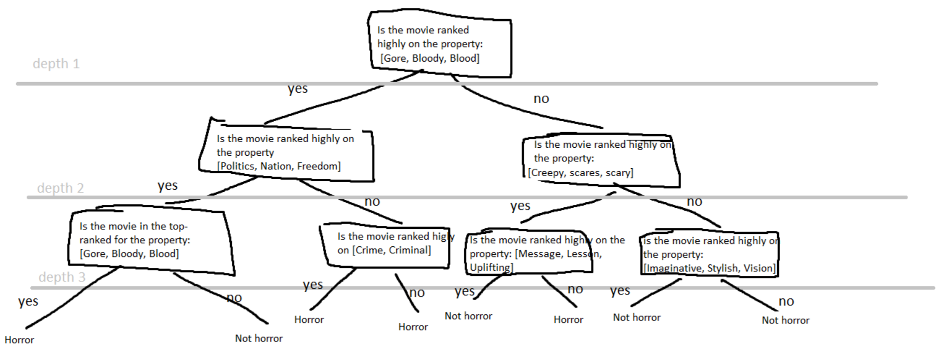

Figure 3.3: This figure shows an example tree from one of our classifiers. Here, we can see that the model increases in complexity as it increases in depth. In this case, we end-up getting better F-score with just a depth-one tree, as the tree begins to overfit at depth three. .

representing salient concepts in the domain, we are also able to offer a similar option; less nodes in the decision tree in exchange for more accuracy.

This chapter continues as follows: We begin by describing the work related to this method, giving valuable context for the utility and potential of our approach. This is followed by an explanation of the method, including the variations we have adopted for our experimental work. We follow this with our qualitative experimentation, explaining how these variations affect the results, as well as the interpretability of the method, and we end with a quantitative analysis on how well we can represent domain knowledge using decision trees constrained to a limited depth.

3.2 Related Work

Linear Classifiers Decision trees, linear SVM's, logistic regression, decision tables, IF Then rules.

What are the available options for interpretable linear classification?

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

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

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

Other Stuff

3.2.1 Interpretable Representations

3.2.2 Interpretable Classifiers

There are two ways in which topic models can be used for document classification. First, a supervised topic model can be used, in which the underlying graphical model is explicitly extended with a variable that represents the class label [3]. Second, the parameters of the multinomial distribution corresponding to a given document can be used as a feature vector for a standard classifier, such as a Support Vector Machine (SVM) or Decision Tree. LDA has been extended by many approaches, e.g. aiming to avoid the need to manually specify the number of topics [**?**], modelling correlations between topics [2], or by incorporating meta-data such as authors [**?**] or time stamps [**?**].

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

3.3 Method

The goal of this method is to obtain a representation composed of salient properties, starting with a domain-specific vector space S_e and its associated bag-of-words (BOW) representation B_w. To obtain these properties, we use a variant of the unsupervised method proposed in [**?**], which we explain in this section.

Rankings entities on words

We can understand that only some words will be properties, as only some correspond to domain knowledge, e.g. in a domain of IMDB movies, the word "the" does not correspond to a property of the domain, but the word "horror" does. Initially, we obtain rankings of entities for each word in the space.

As an initial filtering step, we remove words that do not meet a frequency threshold, with the understanding that words that do not occur in a minimum amount of documents are unlikely to correspond to properties as they are too specific to a subset of movies, which would make them difficult to learn. This leaves us with w_n words. We show the kind of words that would be poorly represented in **??**.

Then, for each considered word w, a logistic regression classifier is trained to find a hyperplane H_w in the space that separates entities e which contain w in their BOW B_e representation from those that do not. The vector v_w perpendicular to this hyperplane is then taken as a direction that models the word w. In **??**, we show an example of this in a toy domain. To rank the objects on the entity, if e is the representation of an entity in the given vector space S_e then we can think of the dot product $v_w \cdot e$ as the value $r_e w$ of object e for vector v_w, and in particular, we take $r_e 1 < r_e 2$ to mean that e_2 has the property labelled with the word w to a greater extent than e_1. The result of this is shown in **??**. Example entities, with their associated highest and lowest ranking properties, are shown in **??**.

Filtering directions to obtain salient properties

With the rankings R_r, we could create a representation of each entity Se, composed of w_n dimensions, where each dimension is a ranking of the entity e on that word $w_r e$. However, many of the words do not properties. In-order to filter these words out, we evaluate them using a scoring metric, and remove the words that are not sufficiently well scored. We use three different metrics:

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

Cohen's Kappa. One problem with accuracy as a scoring function is that these classification problems are often very imbalanced. In particular, for very rare words, a high accuracy might not necessarily imply that the corresponding direction is accurate. For this reason, X proposed to use Cohen's Kappa score instead. In our experiments, however, we found that accuracy sometimes yields better results, so we keep this as an alternative metric.

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

$$p_{wo} = \frac{n(w, o)}{\sum_{w'} \sum_{o'} n(w', o')}$$

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

In principle, we may expect that accuracy and Kappa are best suited for binary features, as they rely on a hard separation in the space between objects that have the word in their BoW representation and those that do not, while NDCG should be better suited for gradual features. In practice, however, we could not find such a clear pattern in the differences between the words chosen by these metrics despite often finding different words. In Table **??**, we show examples of properties scored highly for each domain.

Clustering salient properties

If we consider two directions, "Blood" and "Gore", we can understand both of these to be approximating a property of films; How much blood they contain. Because of this, we can expect their directions to be very similar to each other. Averaging these directions together would result in a direction inbetween them. Similarly, obtaining a hyper plane using a Logistic Regression classifier that uses occurences of both and either of these terms as positive would be similar to this averaged direction. As some entities would have the property of being bloody films, but did not necessarily use the term gore in their reviews, same as some entities having the property but using the term gore not bloody, we can understand that this new hyper plane and associayed direction more accurately represents the property of a bloody film more than either of the terms individually. This is the principle behind our clustering method - going from term directions to property directiona.

A term direction for "beautiful" is nebulous in the sense that we are not necessarily able to intuit its associated property. However, once we cluster the terms to find the property ("beautiful", "cinematography" "shots") we are given valuable context for the word. This is another advantage for clustering, we are able to construct a list of terms that label the property, alllowing us to more easily understand the meaning of thr ranking we induce.

Naturally, it is sometimes not enough to see a list of terms and understand the property without domain knowledge. However, by examining how classifiers use these directions to classify key domain knowledge we are better able to understand what they are modelling. For example, when classifying if a movie is a sci-fi, we see that if a movie is ranked highly on the term "science, scientist", then it is not a sci-fi movie. However, when classifying if a movie is a biography, we see that if a movie is ranked highly on "science, scientist" then it is a biography movie. From this, we can understand that the property is not about mad scientists, but normal non-fiction science.

As this method is sensitive to the first direction selected (if the first direction is not a property then we will likely find a few useless terms before landing on something useful)

Although we are able to find the words that are most salient, the properties in the domain may not correspond directly to words. Further, the properties may not be well described by their associated word. In-order to find better representations of properties, we cluster together similar vectors v_w, following the assumption that those vectors which are similar are representing some property more general than their individual words, and we can find it between them. As the final step, we cluster the best-scoring candidate feature directions v_w. Each of these clusters will then define one of the feature directions to be used in applications. The purpose of this clustering step is three-fold: it will ensure that the feature directions are sufficiently different (e.g. in a space of movies there is little point in having *funny* and *hilarious* as separate features), it will make the features easier to interpret (as a cluster of terms is more descriptive than an individual term), and it will alleviate sparsity issues when we want to relate features with the BoW representation, which will play an important role for the fine-tuning method described in the next section.

As input to the clustering algorithm, we consider the N best-scoring candidate feature directions v_w, where N is a hyperparameter. To cluster these N vectors, we have followed the approach proposed in [**?**], which we found to perform slightly better than K-means. 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. We refer to [**?**] for more details. The output of this step is a set of clusters $C_1, ..., C_K$, where we will identify each cluster C_j with a set of words. We will furthermore write v_{C_j} to denote the centroid of the directions corresponding to the words in the cluster C_j, which can be computed as $v_{C_j} = \frac{1}{|C_j|} \sum_{w_l \in C_j} v_l$ provided that the vectors v_w are all normalized. These centroids $v_{C_1}, ..., v_{C_k}$ are the feature directions that are identified by our method.

Table **??** displays some examples of clusters that have been obtained for three of the datasets that will be used in the experiments, modelling respectively movies, place-types and newsgroup postings. For each dataset, we used the scoring function that led to the best performance on development data(see Section **??**). Only the first four words whose direction is closest to the centroid v_C are shown. **K-Means Derrac's K-Means Variation Mean-shift Hdbscan**

3.3.1 Quantitative Results

We use the data provided by [5], but differ from them in a few ways. First, rankings are done differently (we combine them differently or something?), as well as duplicates being removed from the data. This makes it difficult to directly compare our results to theirs, although they are sometimes similar.

"Second, as the classification problems are heavily imbalanced, most methods are able to achieve a similar accuracy score. Differences between the F1 score, on the other hand, are more pronounced. Overall," [5]

We demonstrate the effectiveness of our approach on five datasets, each with their associated tasks. In table 3.1 we show the vocabulary and document size for each dataset. For the IMDB and place-type spaces, we take them as-is, with the exception of removing empty or duplicated documents. For the other datasets, we remove all terms that do not occur in at least 2 documents, remove all punctuation and convert them to lower case. We retain numbers. The data labelled "After preprocessing" is the

	Data as received		Preprocessed for vector spaces	
Dataset	Vocabulary size	Amt of entities	Vocabulary size	Amt of entiti
IMDB Movies	100,000	15,000	100,000	
Sentiment		50,000		
Placetypes		1383		
Newsgroups		18846		
Reuters				

Table 3.1: We use the preprocessed datasets for the rest of the paper, including to make the vector spaces. This includes removing stopwords, deleting empty spaces, removing punctuation, converting everything to lowercase, and removing terms that do not occur in at least 2 documents..

data used to create the vector spaces.

For our bag-of-words representation, we further filter the corpus by removing terms that do not appear at least (length of the corpus * 0.001) documents. We additionally remove any terms that are in (length of corpus * 0.95) documents. Unlike when finding directions, we are not interested in finding salient properties, rather we simply want to remove noise from the dataset. For some corpuses, this means that we end-up with some empty entities that contained only infrequent terms. We show the vocabulary changes in 3.2.

The classes are also filtered so that any classes without 100 positive instances are removed. One exception is the place-types classes, as these only have a very limited amount of entities to begin with. Additionally, some classes do not contain all documents - we show the stats for all classes in Table .

Place-types and IMDB Movies are both already limited to 100,000 vocabulary terms initially.

- The IMDB Movie Dataset: 15,000 movies represented by aggregated reviews. On the tasks of Movie Genres, 100 IMDB Keywords, and UK + US Age Certific-

	Data as received		Preprocessed for bag-of-words	
Dataset	Vocabulary size	Amt of entities	Vocabulary size	Amt of entiti
IMDB Movies	100,000	15,000	100,000	
Sentiment		50,000		
Placetypes		1383		
Newsgroups		18846		
Reuters				

Table 3.2: This table shows the preprocessing of the datasets that produce the bag-of-words that we use directly on the classifier. In this case, infrequent terms and extremely frequent terms were removed..

	Data as received		Preprocessed	
Dataset	Amt of classes	Amt of entities	Amt of classes	Amt of entities
IMDB Genres				
IMDB Ratings				
IMDB Keywords				
Placetypes Foursquare				
Placetypes OpenCYC				
Placetypes Geonames				
Sentiment	1			
20 Newsgroups				
Reuters				

Table 3.3: Classes vary in the amount of entities they cover for some classes. Additionally, in the preprocessed section we delete classes that do not have at least 100 positive instances..

ates. However, the data made available only gave a mapping for 13978 entities, so we use those instead in this case. As with all datasets, we remove terms that do not occur in at least 13 documents. This resulted in 12 entities left empty, so these entities were also removed, leaving us with 13966 entities. This corpus was

	SVM	DT (N)	DT (3)	DT (2)	DT (1)
PPMI	0.594	**0.441**	**0.44**	**0.441**	**0.315**
PCA 50	0.509	0.418	0.308	0.418	0.229
PCA 100	0.577	0.412	0.36	0.412	0.238
PCA 200	0.597	0.409	0.342	0.409	0.24
D2V 50	0.592	0.308	0.308	0.308	0.244
D2V 100	0.613	0.335	0.324	0.335	0.234
D2V 200	**0.619**	0.369	0.369	0.369	0.251
AWV 50	0.348	0.233	0.233	0.233	0.213
AWV 100	0.378	0.236	0.236	0.236	0.208
AWV 200	0.451	0.236	0.236	0.236	0.22
MDS 50	0.381	0.242	0.242	0.242	0.191
MDS 100	0.432	0.238	0.238	0.238	0.148
MDS 200	0.481	0.243	0.243	0.243	0.188

Table 3.4: Results for 20 newsgroups.

already limited to only contain 100,000 vocabulary terms. As with all datasets, we remove all terms that are not included in at least 2 entities.

- Flickr Place-types: 1,383 place-types. On the tasks of three different place-types, Foursquare, Geonames and OpenCYC.
- The 20-Newsgroups dataset: 18,846 newsgroup posting in 20 different categories. On the task of identifying which of the 20 categories the posting is from.
- The IMDB Sentiment Dataset: 50,000 movie reviews, with binary tags for either positive or negative. On the task of identifying if the review is positive or negative.
- The Reuters Dataset: 10655 News articles. On the task of identifying the category of the article.

To test the ability of the identified directions to accurately represent domain concepts in a ranking, we use low-depth decision-trees. Although these classifiers are not intended to be competitive with more complex classifiers like unbounded decision trees or SVM's, we find that our rankings are sometimes able to outperform these approaches using only a single decision node (equivalent to finding the best cutoff in a single ranking for classification). We use the F1 metric for our experiments, as almost all classes in each dataset are unbalanced.

We obtain the unsupervised representations as follows:

- For the averaged word-vectors (AWV) and the weighted averaged word vectors (AWVw), we average the glove 6B word-vectors[1] obtained from the Wikipedia 2014 + Gigaword 5 corpuses. As these are only available in size 50, 100 and 200, and there are not many other commonly used pre-trained word-vectors that offer multiple dimension sizes, differing from other methods we only obtain AWV and AWVw representations of size 50, 100 and 200. As these dimension sizes are hyper-parameters, we can consider average word vectors to be disadvantaged on some tasks, but as it is unlikely that there is too much benefit in training our own word-vectors from the relatively small domains, we opted to simplify the process and simply remove this as a hyper-parameter for this method, as well as the averaged method. The averaged word vectors are obtained by multiplying the vectors by the PPMI values, and finding the weighted average of all vectors multiplied in this way. We obtain size 50, 100 and 200 dimensional spaces for all other space-types to keep it consistent with AWV.
- PPMI (Put it above)
- We obtain the MDS spaces for the movies, place-types and wines datasets from the data made available by [5], to obtain the MDS spaces for the other datasets, we use the same method as [5] and using default parameters for the MDSJ

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

library. For all domains apart from sentiment, we obtain 50, 100 and 200 dimensional spaces. For the sentiment domain, we do not obtain an MDS space due to memory constraints (as it has 50,000 docs). This is a limitation of classic MDS.

- So that we can use the sparse PPMI matrices when obtaining the space, we the TruncatedSVD method from scikit-learn method[2] with default parameters. For each domain, we obtain 50, 100 and 200 dimensional spaces.

- For the Doc2Vec vectors, we use hyperparameter optimization to select the appropriate parameters, as the quality of the end space is typically reliant on well-tuned hyperparameters for the dataset. We use [9] as a guideline for which parameters to optimize, re-using the parameters that stayed constant for both their datasets in their tests, specificially the dbow method, glove6B pre-trained 300-dimensional word-vectors, training those word vectors while training the representation, a sub-sampling of 10(-5), and a negative sample of 5. We tune and select between the values of the window size 5, 15, 50, the minimum frequency count 1, 5, 20, and epoch size 20, 40, 100, and as in the other methods, we obtain vectors of size 50, 100, 200, but the hyperparameters for each of these is found individually, as the different space sizes are later evaluated on how well they can produce good directions. We evaluate the quality of the space using a Gaussian SVM on a selected task for each dataset, in the case of Reuters, Newsgroups and Sentiment, we use their associated tasks, for Movies we use the Genres task and Place-types the Foursquare task, as these tasks represent essential concepts in the domain.

Table 1 shows how well unsupervised representations perform. Topic models are included to demonstrate the difference between other simple and interpretable approaches, and Random Forest's are included to demonstrate the difference between our simple but interpretable approach and a model that typically performs well at the task [6], but is difficult to interpret.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.htm

Table 2 demonstrates the difference between unsupervised representations and salient properties, and Table 3 demonstrates the difference between salient properties and clustered salient properties.

3.3.2 Interpretability Results

Chapter 4

Fine-tuning Vector Spaces to Improve Their Directions

"Commonly, these representations are made in a single vector space with similarity being the main structure of interest. However, recent work by Mikolov et al. (2013b) on a word-analogy task suggests that such spaces may have further use- ful internal regularities. They found that seman- tic differences, such as between big and small, and also syntactic differences, as between big and bigger, were encoded consistently across their space. In particular, they solved the word-analogy problems by exploiting the fact that equivalent re- lations tended to correspond to parallel vector- differences. [10]

[10] "Explicitly designing such structure into a neural network model results in representations 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 de- composition can produce substantial im- provements on semantic-similarity, pos- induction and word-analogy tasks."

4.1 Experiments

We find that non-linearity is useful.

Chapter 5

Investigating Neural Networks In Terms Of Directions

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

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

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

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

[5] 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.

[6] Manuel Fern and Eva Cernadas. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems ? 15:3133–3181, 2014.

[7] Peter Gärdenfors. Conceptual spaces. (September), 2014.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

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

[14] Michael Veale. Logics and practices of transparency and opacity in real-world applications of public sector machine learning. 2017.