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 .....

Copyright © 2011 Name Lastname.

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”.

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

Abstract **iv**

Acknowledgements **v**

Contents **vi**

List of Publications **ix**

List of Figures **x**

List of Tables **xi**

List of Algorithms **xiv**

List of Acronyms **xv**

0.0.1 Definitions xv

1 Introduction **1**

1.1 Motivation 1

1.1.1 Directions 3

1.2 Interpretability 4

1.3 Thesis Overview / Contributions 5

2 Background **6**

2.1 Text 6

2.1.1 Bag-of-words 6

2.2 Text classification 6

2.2.1 Decision Trees 6

2.2.2 Support Vector Machines 7

2.2.3 Neural Networks 7

2.2.4 Semantic Spaces 7

2.2.5 Document Representations 8

2.3 Interpretable Representations 8

3 Converting Vector Spaces into Interpretable Representations **10**

3.1 Introduction 10

3.1.1 Semantic Relations 10

3.1.2 Producing an Interpretable Property Representation 12

3.1.3 Using a Property Representation in a Linear Classifier 13

3.2 Related Work 15

3.2.1 Semantic Relations & Their Applications 15

3.2.2 Interpretable Representations 15

3.2.3 Interpretable Classifiers 15

3.3 Method 16

3.3.1 Corpus Preprocessing 17

3.3.2 Term Directions and Rankings 17

3.3.3 Filtering Terms 18

3.3.4 Feature-Rankings 18

3.3.5 Quantitative Results 22

3.3.6 Interpretability Results 23

4 Fine-tuning Vector Spaces to Improve Their Directions **39**

4.1 Experiments 39

5 Investigating Neural Networks In Terms Of Directions **40**

GNU Free Documentation License **41**

Bibliography **50**

List of Publications

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

-
-

List of Figures

1.1 Bag-of-words 2

1.2 Example properties 2

3.1 Movies and selected associated dimensions, and their use in a linear classifier. 11

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. 13

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. 14

3.4 Original And Converted. 17

3.5 A conceptual space of movies, where regions correspond to properties and entities are points. 22

List of Tables

3.1 Two of the following entities: Those classified as horror, those classified as horror and romance, and those classified as romance with their associated highest value PPMI terms. We show the highest positive instances here as the representation is sparse, even though we can also expect the terms that are low scoring to be similar too. 23

3.2 Results for 20 newsgroups (Original table F1 scores only). 24

3.3 Results for 20 newsgroups SVM. 25

3.4 Results for 20 newsgroups DT3. 25

3.5 Results for 20 newsgroups DT2. 26

3.6 Results for 20 newsgroups DT1. 26

3.7 Results for 20 newsgroups DTN. 27

3.8 Results for Sentiment SVM. 27

3.9 Results for Sentiment DT1. 28

3.10 Results for Sentiment DT2. 28

3.11 Results for Sentiment DT3. 29

3.12 Results for Sentiment DTN. 29

3.13 Results for OpenCYC SVM. 30

3.14 Results for OpenCYC DT3. 30

3.15 Results for OpenCYC DT2. 31

3.16 Results for OpenCYC DT1. 31

3.17 Results for OpenCYC DTN. 32

3.18 Results for Foursquare SVM. 32

3.19 Results for Foursquare DT3. 33

3.20 Results for Foursquare DT2. 33

3.21 Results for Foursquare DT1. 34

3.22 Results for Foursquare DTN. 34

3.23 Results for Geonames SVM. 35

3.24 Results for Geonames DT3. 35

3.25 Results for Geonames DT2. 36

3.26 Results for Geonames DT1. 36

3.27 Results for Geonames DTN. 37

3.28 Results for Genres SVM. 37

3.29 Results for Genres DT3. 37

3.30 Results for Genres DT2. 37

3.31 Results for Genres DT1. 37

3.32 Results for Genres DTN. 37

3.33 Results for Keywords SVM. 37

3.34 Results for Keywords DT3. 37

3.35 Results for Keywords DT2. 37

3.36 Results for Keywords DT1. 37

3.37 Results for Keywords DTN. 37

3.38 Results for Ratings SVM. 37

3.39 Results for Ratings DT3. 37

3.40 Results for Ratings DT2. 37

3.41 Results for Ratings DT1. 38

3.42 Results for Ratings DTN. 38

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.

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

Vector Space A representation composed of vectors.

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

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

Bag-Of-Words a 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.

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

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

Cluster label A cluster of words that describe a property.

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

Feature direction A distinct direction in the space that corresponds to a useful feature in the domain.

Feature rankings The rankings induced from a feature vector.

Chapter 1

Introduction

1.1 Motivation

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

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

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

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

Entity: X

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

Entity: Y

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

Entity: Z

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

Figure 1.1: Bag-of-words

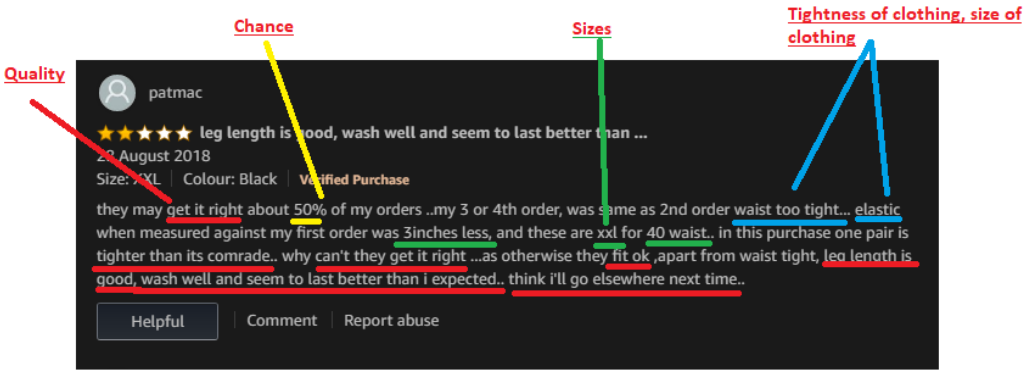

Figure 1.2: Example properties

on the metadata available from the site itself, but for the latter the way to obtain this information is less clear. Although we may infer that the rating has some indication of these properties, it does not describe the properties or the degree to which the review refers to them. This kind of information is valuable for making sense of the world

of unstructured text, and has broad applications, e.g. The most immediate example is perhaps that they allow for a natural way to implement critique-based recommendation systems, where users can specify how their desired result should relate to a given set of suggestions [**?**]. For instance, [**?**] propose a movie recommendation system in which the user can specify that they want to see suggestions for movies that are "similar to this one, but scarier". If the property of being scary is adequately modelled as a direction in a semantic space of movies, such critiques can be addressed in a straightforward way. Similarly, in [**?**] a system was developed that can find "shoes like these but shinier", based on a semantic space representation that was derived from visual features. Semantic search systems can use such directions to interpret queries involving gradual and possibly ill-defined features, such as "*popular* holiday destinations in Europe" [**?**]. While features such as popularity are typically not encoded in traditional knowledge bases, they can often be represented as semantic space directions.

1.1.1 Directions

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

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

we are able to learn simple linear classifiers that perform well.

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

1.2 Interpretability

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

Semantic relationships can be obtained from semantic spaces.

applications/need for good interpretability:

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

properties of an interpretable classifier:

- Complexity: 'the magic number is seven plus or minus two' [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

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

Then, we count the occurrences of each word

- Frequency
- Tf-idf
- PPMI

2.2 Text classification

2.2.1 Decision Trees

- Explanation of what decision trees are

- Explanation that they may not perform well on sparse information

2.2.2 Support Vector Machines

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

2.2.3 Neural Networks

- Difference between SVM and Nnet

2.2.4 Semantic Spaces

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

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

Vector spaces are a popular way to represent unstructured text data, and have been broadly applied to and transformed by supervised approaches. They vary in method, producing structure from Cosine Similarity, Matrix Factorization, Word-Vectors/Doc2Vec, etc. They also vary in how they linearly separate entities. However, their commonality is that they are able to represent semantic relationships spatially. See Section 2.2.4 This brings up an essential point: When using a semantic space, are we taking advantage of relationships that are discriminative or incorrect? The danger of relying on these

spaces and the models that use them has greatly affected their adoption in critical application areas like medicine, and has raised legal concerns about their application in e.g. determining if someone is suitable for a loan.

See Section 2.2.4

- Word-vectors

2.2.5 Document Representations

Conceptual spaces

- PCA
- MDS

2.3 Interpretable Representations

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

[**?**] Sparse PCA (Why not compare lol)

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

combine topic models and vector space models. For example, the Gaussian LDA model represents topics as multivariate Gaussian distributions over a word embedding [**?**]. Beyond document representation, topic models have also been used to improve word embedding models, by learning a different vector for each topic-word combination [**?**].

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

Chapter 3

Converting Vector Spaces into Interpretable Representations

3.1 Introduction

This chapter describes how to go from a domain-specific Semantic Space (see Section 2.2.4) to producing linear classifiers driven by features where each dimension is a ranking on a distinct labelled property in the domain. By a property, we mean a semantically meaningful aspect of domain knowledge that entities can be ranked on. We show an example representation and classifier that were obtained from a Semantic Space constructed from IMDB movie reviews in figure 3.1. We find that these representations are more interpretable than standard baseline interpretable representations (e.g. Topic Models), and also find that they outperform baseline harder to understand classifiers (Linear SVM's) in F1-score for some tasks, even when constrained to a very simple linear classifier.

3.1.1 Semantic Relations

The success of these semantic spaces similarity-based structure has lead many to investigate how to obtain formal relationships from them. One such striking example is in that of linear analogies in word-vectors (see Section **??**, where it was found that the

and rep.png

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

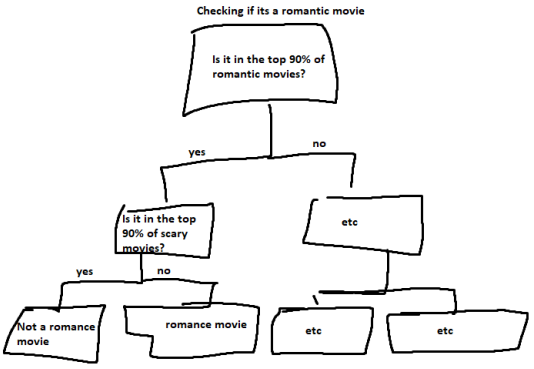

Figure 3.1: Movies and selected associated dimensions, and their use in a linear classifier..

vector XXXX[]King queen blah blah[]XXXXXXX, formally justified in [**?**]. These relationships have been expanded on, for example [13] found that "equivalent relations tended to correspond to parallel 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. Additionally, [**?**] found that word distances between gendered words (e.g. male, female, she, her) and occupational words e.g. (nurse, programmer) were correlated to the percentage of occupation that gender had for that role in different time periods.

The semantic relation that we focus in on this paper are directions that correspond to salient features from the considered domain. A "direction" refers in this case to 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 directions indicating lexical features such as the frequency of occurrence and polarity of words.

3.1.2 Producing an Interpretable Property Representation

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.

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.

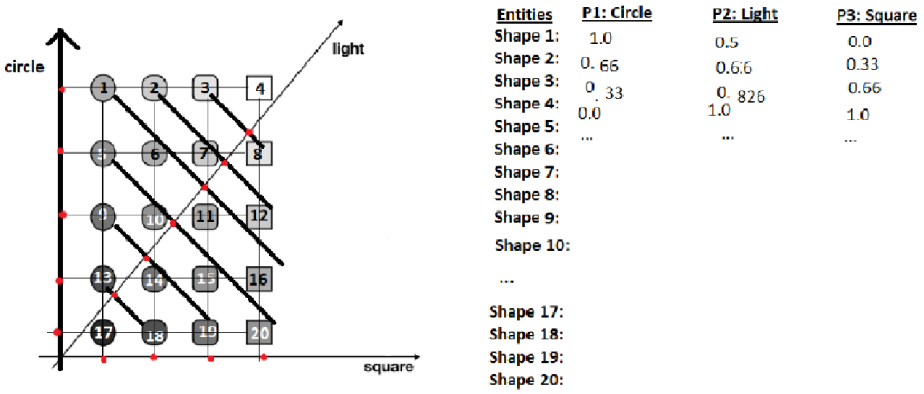

Figure 3.2: This figure shows a 2d toy space where entities are shapes and directions are properties. We demonstrate on the right the method to induce a ranking from the directions, in particular by using the dot-product of the entity point on the directions vector. In the same way for a more complex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

3.1.3 Using a Property Representation in a Linear Classifier

Our method can use any vector space that linearly separates entities, and so it has potential longevity. This means that our method is relying on structure in the space that does not directly correspond to our desired representation - we can view our approach as a linear transformation of the space. We address this problem in Chapter **??**. However, we have the capability to leverage many different methods to construct a vector space for our representation, so as long as dense representations of entities exist it will be possible to use our method, and as they are improved the results that our method can achieve will be improved too. This kind of flexibility also gives us the potential to combine the resultant representations from different vector spaces for classification, e.g. concatenating the vectors from different spaces.

Topic models such as Latent Dirichlet Allocation (LDA) represent documents as multinomial distributions over latent topics, where each of these topics corresponds to a

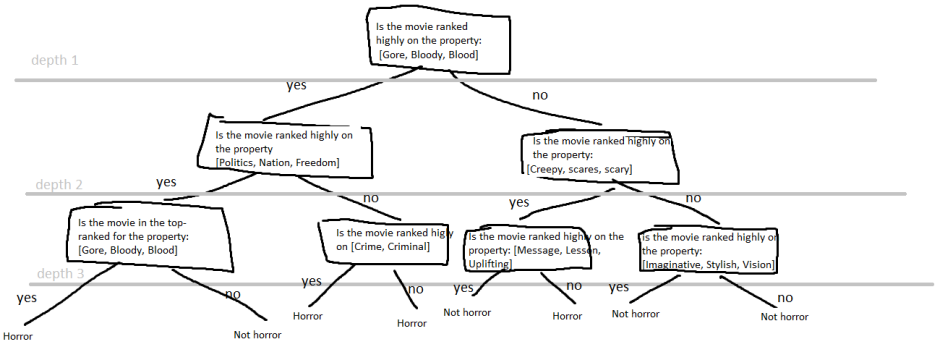

Figure 3.3: This figure shows an example tree from one of our classifiers. Here, we can see that the model increases in complexity as it increases in depth. In this case, we end-up getting better F-score with just a depth-one tree, as the tree begins to overfit at depth three. .

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.

This chapter continues as follows: We begin by describing the work related to this , 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

3.2.1 Semantic Relations & Their Applications

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

What are the available options for interpretable linear classification?

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

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

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

Other Stuff

3.2.2 Interpretable Representations

There are two ways in which topic models can be used for document classification. First, a supervised topic model can be used, in which the underlying graphical model is explicitly extended with a variable that represents the class label [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

This section details the methodology to go from a domain-specific Bag-Of-Words 2.1.1 and Semantic Space 2.2.4 constructed from text documents to an interpretable representation of feature-rankings. To begin, we filter out terms that will not be well represented in the space, and in-turn will not result in good feature rankings. We do so by first removing all terms below a frequency threshold then, we use various metrics to directly evaluate the representation of each term in the space and remove those that are poorly represented. We can consider the terms remaining to be candidate feature-directions, as they are all well represented in the space. However, it is possible that they represent similar directions despite being different terms, e.g. "blood" and "bloody". Additionally, we can understand that a good feature will not correspond directly to a term, but instead to a concept. In order to obtain good and unique features, we cluster together similar feature-directions to obtain a final representation of feature-directions and feature-rankings, labelled by the terms that have been clustered together. We show examples of this final representation in **??**.

We start 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.

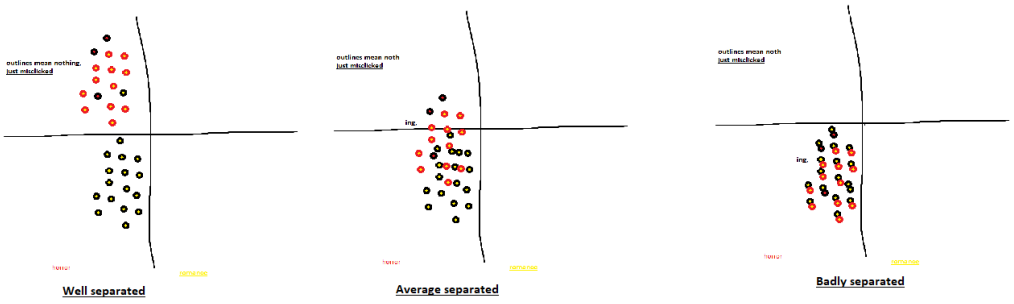

Figure 3.4: Original And Converted.

3.3.1 Corpus Preprocessing

We begin with an unstructured corpus of text documents C_D from a domain, for example a collection of IMDB reviews, where each document contains the concatenated reviews for that movie. From these reviews, we want to extract terms that could correspond to good feature-rankings. To do so, we convert all text to lower-case, remove all punctuation, remove unnecessary whitespace, and remove all non-alphanumeric characters like emojis. From here, we remove standard English stop words using the NLTK library [**?**]. We show an example of a review's original and converted formats in Figure 3.4.

Once we have extracted these terms, we obtain a Bag-Of-Words where we count the frequency of each term, see 2.1.1. Then, we remove all words that do not occur in f documents. This filtered Bag-Of-Words $FBOW$ is used for the rest of the method.

3.3.2 Term Directions and Rankings

Then, for each considered word w, a Support Vector Machine **??** 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. This hyper-plane also considers the degree to which an document is relevant for a term in the sense that those close to the hyper-plane on the positive side are less relevant than those furthest from the hyper-

plane (and those furthest on the negative side are the least relevant for the property). Following this insight, we can consider the vector v_w perpendicular to this hyperplane as a direction that models documents from least relevant to most relevant for 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 **??**.

3.3.3 Filtering Terms

3.3.4 Feature-Rankings

Obtaining Term Directions

Filtering Directions

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 $\mathit{ppmi}(w, e)$, of the word w in the BoW representation of entity e where $\mathit{ppmi}(w, e) = \max\left(0, \log\left(\frac{p_{we}}{p_{w*} \cdot p_{*o}}\right)\right)$, 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.

Obtaining 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**

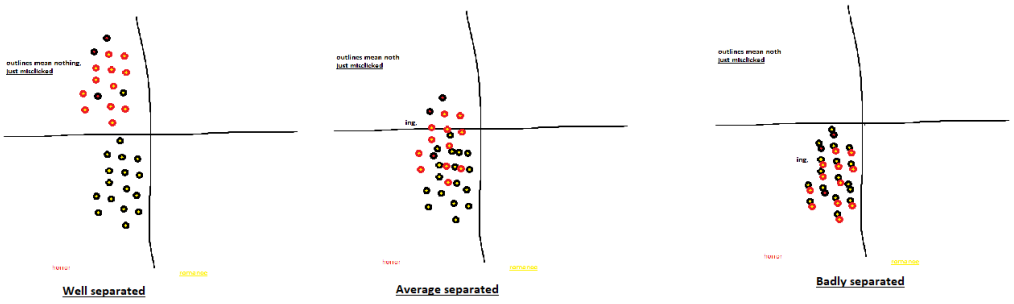

Figure 3.5: A conceptual space of movies, where regions correspond to properties and entities are points..

Variation Mean-shift Hdbscan

3.3.5 Quantitative Results

This section focuses on using linear classifiers to determine how well our method represents domain knowledge compared to standard baselines. We can understand that an accurate representation of domain knowledge will be one that ensures semantically distinct entities are separated, and semantically similar entities are close together. Put another way, if the space is representing domain knowledge well we can expect that the space should be linearly separable for key semantics of the domain. For example, a good vector space in the domain of movies constructed from IMDB movie reviews should contain a natural separation of entities into genres, where Horror movies are spatially distant from Romance movies, and movies that are Romantic Horrors would be somewhere inbetween. We can see an example in Figure 3.5. For a Bag-Of-Words, we can expect similar entities to have similarly scoring terms **??**.

When selecting the parameters to use for the doc2vec space when obtaining directions, we choose the one that scored the highest for its class on a Linear SVM, rather than tuning the entire process around the doc2vecs vectors. We use the kind of multi-class strategy as a hyper-parameter for each class-type in the grid search. We test the

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

Table 3.1: Two of the following entities: Those classified as horror, those classified as horror and romance, and those classified as romance with their associated highest value PPMI terms. We show the highest positive instances here as the representation is sparse, even though we can also expect the terms that are low scoring to be similar too..

OneVsOne classifier method, treating each as binary problems, the OneVsRest method and the OutputCode method. We are not able to obtain an MDS space for sentiment or doc2vec spaces for placetypes/movies.

As our work performs well even at lower-depth trees, this gives potential users more flexibility in how they want to present the information, e.g. to a potential client. Compared to bag-of-words, which loses its representation capabilities the lower the depth.

Results for vector spaces and bag-of-words

3.3.6 Interpretability Results

	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 3.2: Results for 20 newsgroups (Original table F1 scores only).

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.96490308	0.492247891	0.457299224	0.711658333	0.376247516
AWV 50	0.836962294	0.348019806	0.287991958	0.221993012	0.805053757
MDS 50	0.850165959	0.380514451	0.319361959	0.249390206	0.80239975
D2V 50	0.969251195	0.583050235	0.552867243	0.762328571	0.472039711
PCA 100	0.96798991	0.573312826	0.548472358	0.76084381	0.45994634
AWV 100	0.8581718	0.37842844	0.32249064	0.247582046	0.802600456
MDS 100	0.877734997	0.431782978	0.367431954	0.304064453	0.744501764
D2V 100	0.969576474	0.607208735	0.580914149	0.763888161	0.503862585
PCA 200	0.96814923	0.600195646	0.579020555	0.736568432	0.506431769
AWV 200	0.960010621	0.451033454	0.417523124	0.628312372	0.351778753
MDS 200	0.964630908	0.480837639	0.450593914	0.789387894	0.345709284
D2V 200	0.969138343	0.619296679	0.597329654	0.731193811	0.537102079

Table 3.3: Results for 20 newsgroups SVM.

	F1	PREC	RECALL
PCA 50	0.328304254	0.206683241	0.797709964
AWV 50	0.232797129	0.136574918	0.787908695
MDS 50	0.242237348	0.144618794	0.745359468
D2V 50	0.262485129	0.159552724	0.739667629
PCA 100	0.348355036	0.590710997	0.247011539
AWV 100	0.236469254	0.139836252	0.765381631
MDS 100	0.237535645	0.140254022	0.775272724
D2V 100	0.241880243	0.144911904	0.731095552
PCA 200	0.360425936	0.540033963	0.270470942
AWV 200	0.235578784	0.139287267	0.763171354
MDS 200	0.243002892	0.145997046	0.72416485
D2V 200	0.242582271	0.144063637	0.767313249

Table 3.4: Results for 20 newsgroups DT3.

	F1	PREC	RECALL
PCA 50	0.351456416	0.23188865	0.725588605
AWV 50	0.277555185	0.175212293	0.667373129
MDS 50	0.262512227	0.160257903	0.725294194
D2V 50	0.291764949	0.1910914	0.616623507
PCA 100	0.36290335	0.24352329	0.711882698
AWV 100	0.260604694	0.160646408	0.689841882
MDS 100	0.262095206	0.161124391	0.70203524
D2V 100	0.275263826	0.173804325	0.661305683
PCA 200	0.359230759	0.239547295	0.717920958
AWV 200	0.249564678	0.151789776	0.701313625
MDS 200	0.253938183	0.153568133	0.733049213
D2V 200	0.273966602	0.175932631	0.618748321

Table 3.5: Results for 20 newsgroups DT2.

	F1	PREC	RECALL
PCA 50	0.237796799	0.139329768	0.810816271
AWV 50	0.212916931	0.123466917	0.772798069
MDS 50	0.174946697	0.099635471	0.716605845
D2V 50	0.206890545	0.119473856	0.771060354
PCA 100	0.241437752	0.142104023	0.80217534
AWV 100	0.208197142	0.1202674	0.774308646
MDS 100	0.186613434	0.105176833	0.826757278
D2V 100	0.196855298	0.112987628	0.76381335
PCA 200	0.24018672	0.140977055	0.810699841
AWV 200	0.220189174	0.128863697	0.755880029
MDS 200	0.187525181	0.105846321	0.821304769
D2V 200	0.207592806	0.121252635	0.72097962

Table 3.6: Results for 20 newsgroups DT1.

	F1	PREC	RECALL
PCA 50	0.41928432	0.405469685	0.434073508
AWV 50	0.30215353	0.281684719	0.325830202
MDS 50	0.293534597	0.279429073	0.309139919
D2V 50	0.38212279	0.396860235	0.368440707
PCA 100	0.41444605	0.39177829	0.439897947
AWV 100	0.296573651	0.277583826	0.318352508
MDS 100	0.303315524	0.284560799	0.324716839
D2V 100	0.369399726	0.390914592	0.350129557
PCA 200	0.405630591	0.384607892	0.429084378
AWV 200	0.307071322	0.287564233	0.329417555
MDS 200	0.295333334	0.275822911	0.31781401
D2V 200	0.350219973	0.37188488	0.330940371

Table 3.7: Results for 20 newsgroups DTN.

	F1	Precision	Recall
PCA 50	0.886155063	0.876447574	0.89608
AWV 50	0.755943328	0.756367131	0.75552
D2V 50	0.86072602	0.862176914	0.85928
PCA 100	0.891788242	0.881575862	0.90224
AWV 100	0.801005145	0.798743139	0.80328
D2V 100	0.872537705	0.8680814	0.87704
PCA 200	0.892751333	0.881779165	0.904
AWV 200	0.828502127	0.823622421	0.83344
D2V 200	0.878093492	0.874851108	0.88136

Table 3.8: Results for Sentiment SVM.

	avg_f1	avg_prec	avg_recall
PPMI	0.699599048	0.574907331	0.89336
PCA 50	0.124525871	0.590631365	0.0696
AWV 50	0.495850524	0.608827297	0.41824
D2V 50	0.664416948	0.625264644	0.7088
PCA 100	0.705496839	0.8337515	0.61144
AWV 100	0.599821349	0.678968655	0.5372
D2V 100	0.629258374	0.567823344	0.7056
PCA 200	0.705458908	0.833496892	0.61152
AWV 200	0.651572425	0.63521009	0.6688
D2V 200	0.663735313	0.603961695	0.73664

Table 3.9: Results for Sentiment DT1.

	F1	PREC	RECALL
PPMI	0.718834534	0.606500577	0.88224
PCA 50	0.769670528	0.72629188	0.81856
AWV 50	0.621045392	0.594426565	0.65016
D2V 50	0.708094413	0.606995885	0.8496
PCA 100	0.769658907	0.726208218	0.81864
AWV 100	0.693895748	0.607262905	0.80936
D2V 100	0.707420276	0.620556461	0.82256
PCA 200	0.769572084	0.72605364	0.81864
AWV 200	0.663435963	0.609166945	0.72832
D2V 200	0.694119925	0.581737531	0.86032

Table 3.10: Results for Sentiment DT2.

	F1	PREC	RECALL
PPMI	0.730120722	0.624793671	0.87816
PCA 50	0.773241796	0.790326623	0.75688
AWV 50	0.716599792	0.668978912	0.77152
D2V 50	0.706911296	0.687509451	0.72744
PCA 100	0.773228733	0.790212126	0.75696
AWV 100	0.651385973	0.701226824	0.60816
D2V 100	0.724228567	0.658482289	0.80456
PCA 200	0.773133962	0.790014194	0.75696
AWV 200	0.686097207	0.692470665	0.67984
D2V 200	0.688555129	0.668651541	0.70968

Table 3.11: Results for Sentiment DT3.

	F1	PREC	RECALL
PPMI	0.709744995	0.714842165	0.70472
PCA 50	0.778979658	0.786086673	0.772
AWV 50	0.645669922	0.645980141	0.64536
D2V 50	0.699698128	0.704090725	0.69536
PCA 100	0.773306533	0.777212121	0.76944
AWV 100	0.647586262	0.648053197	0.64712
D2V 100	0.683666547	0.682060674	0.68528
PCA 200	0.770502991	0.773265651	0.76776
AWV 200	0.663097363	0.657491152	0.6688
D2V 200	0.665625	0.666693419	0.66456

Table 3.12: Results for Sentiment DTN.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.803602223	0.443589681	0.308199143	0.339736332	0.638891067
AWV 50	0.810116881	0.457499286	0.323892593	0.349551692	0.661907853
MDS 50	0.813565817	0.457003603	0.323002846	0.364634555	0.61204707
PCA 100	0.821421728	0.450122408	0.32121053	0.35527646	0.614052074
AWV 100	0.835409082	0.455127069	0.336542753	0.364636358	0.6053566
MDS 100	0.83713355	0.474728943	0.352474877	0.393159213	0.599006284
PCA 200	0.84671393	0.474074812	0.360353916	0.395372385	0.591897289
AWV 200	0.849971259	0.465697168	0.355000122	0.385607296	0.587777361
MDS 200	0.861084499	0.476306142	0.367476644	0.425416709	0.541024925

Table 3.13: Results for OpenCYC SVM.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.695152328	0.341785127	0.157118987	0.242841085	0.576797613
AWV 50	0.728300441	0.39586446	0.226931085	0.281210223	0.668370655
MDS 50	0.728492048	0.361747133	0.19053657	0.26813119	0.555800661
PCA 100	0.646100786	0.305716926	0.103702685	0.214560016	0.531547629
AWV 100	0.69898448	0.352968414	0.174088352	0.251372807	0.592390968
MDS 100	0.722552213	0.355478522	0.17636028	0.262349467	0.551113186
PCA 200	0.715845947	0.333121225	0.153256681	0.245505135	0.517977489
AWV 200	0.760682123	0.382249254	0.221321346	0.282220454	0.592115806
MDS 200	0.730599732	0.374349802	0.195944453	0.279258101	0.567639997

Table 3.14: Results for OpenCYC DT3.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.707606821	0.343429484	0.156929616	0.251027506	0.543482807
AWV 50	0.650890975	0.375734032	0.183056015	0.256467865	0.702350672
MDS 50	0.707415214	0.387617362	0.194958449	0.275974626	0.650954337
PCA 100	0.623107875	0.310355155	0.102113833	0.208258091	0.608829572
AWV 100	0.689020885	0.366415742	0.181031062	0.258842906	0.626985335
MDS 100	0.694002683	0.369986155	0.180288011	0.265420819	0.610498481
PCA 200	0.689020885	0.333046463	0.147294515	0.245169324	0.519115278
AWV 200	0.646292393	0.325676238	0.128022927	0.234108504	0.534889609
MDS 200	0.700134125	0.397197263	0.204265502	0.294908338	0.608125541

Table 3.15: Results for OpenCYC DT2.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.553745928	0.310633902	0.092680202	0.206504902	0.626587132
AWV 50	0.624640736	0.382961496	0.17878069	0.262022566	0.711241624
MDS 50	0.615443572	0.35823563	0.146293722	0.240950353	0.697990127
PCA 100	0.585744395	0.346181476	0.13321519	0.228755781	0.711317277
AWV 100	0.620042154	0.380088459	0.176711823	0.254076387	0.754085962
MDS 100	0.624065913	0.364362146	0.155358843	0.245217405	0.708701922
PCA 200	0.601073002	0.344834471	0.134633794	0.229762238	0.690818169
AWV 200	0.643993102	0.349374827	0.146997089	0.236040865	0.672062735
MDS 200	0.597049243	0.358243938	0.143631945	0.237385589	0.729803158

Table 3.16: Results for OpenCYC DT1.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.831960146	0.297281859	0.174312986	0.284553476	0.311202271
AWV 50	0.843839816	0.361596554	0.244157721	0.335722254	0.391792196
MDS 50	0.842690171	0.305261263	0.187421961	0.319838245	0.291955087
PCA 100	0.831768538	0.308890047	0.18386471	0.311999386	0.30584207
AWV 100	0.825828703	0.3157161	0.187815304	0.299946267	0.333236166
MDS 100	0.826403526	0.29890705	0.170454934	0.273414301	0.329642427
PCA 200	0.829852462	0.283509252	0.157446109	0.285790467	0.281264167
AWV 200	0.825062272	0.315853073	0.187303114	0.28455479	0.354887247
MDS 200	0.826211918	0.257117269	0.118759787	0.273406638	0.242659775

Table 3.17: Results for OpenCYC DTN.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.895572264	0.568015809	0.501678075	0.487221026	0.680933505
AWV 50	0.932330827	0.54620429	0.499671391	0.626786958	0.483981376
MDS 50	0.908103592	0.586608962	0.515425669	0.512247438	0.686226545
PCA 100	0.923141186	0.558970742	0.491390942	0.619275607	0.509368588
AWV 100	0.94235589	0.594804035	0.555653205	0.668948188	0.53545577
MDS 100	0.926482874	0.615292664	0.557389577	0.580419711	0.654623981
PCA 200	0.917293233	0.541635593	0.475003669	0.618051665	0.482036476
AWV 200	0.923141186	0.621747387	0.572616698	0.588166472	0.659395043
MDS 200	0.932330827	0.619167077	0.553015718	0.617480101	0.620863297

Table 3.18: Results for Foursquare SVM.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.839598997	0.372257009	0.267318657	0.30580459	0.475608395
AWV 50	0.888888889	0.333089916	0.248206274	0.405456349	0.282643327
MDS 50	0.830409357	0.419208689	0.316494416	0.342341941	0.540588045
PCA 100	0.868838764	0.258742227	0.169002156	0.285630499	0.236480745
AWV 100	0.850459482	0.452009868	0.353271904	0.375570892	0.567514666
MDS 100	0.898913952	0.420948186	0.345823805	0.424938272	0.417032335
PCA 200	0.860484545	0.388032162	0.296331387	0.327684978	0.475624149
AWV 200	0.83625731	0.398606427	0.303172463	0.311572257	0.553111974
MDS 200	0.859649123	0.453720532	0.367624409	0.396116154	0.530929926

Table 3.19: Results for Foursquare DT3.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.82289056	0.392580599	0.281920837	0.309401801	0.536926586
AWV 50	0.827903091	0.478278996	0.381606764	0.36842967	0.681460044
MDS 50	0.822055138	0.424355448	0.308763351	0.361969434	0.512724306
PCA 100	0.755221387	0.296725552	0.167952229	0.209764361	0.50684722
AWV 100	0.904761905	0.389295248	0.325049447	0.490226337	0.322829039
MDS 100	0.897243108	0.409821491	0.332196168	0.46487494	0.366426887
PCA 200	0.730994152	0.307534481	0.177887091	0.232183759	0.455289988
AWV 200	0.857978279	0.468783935	0.385734234	0.391391941	0.584325748
MDS 200	0.803675856	0.42673129	0.312371899	0.342173461	0.56679838

Table 3.20: Results for Foursquare DT2.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.662489557	0.229077055	0.079211523	0.147644224	0.510816983
AWV 50	0.722639933	0.396225837	0.275700337	0.282063751	0.665633342
MDS 50	0.521303258	0.260605438	0.090701456	0.161991543	0.66610041
PCA 100	0.746867168	0.323216815	0.186288095	0.231022684	0.537860651
AWV 100	0.766917293	0.401290958	0.288812617	0.277038046	0.727641889
MDS 100	0.914786967	0.437858477	0.374376011	0.500864198	0.388933027
PCA 200	0.730994152	0.342263034	0.202224854	0.240331095	0.594342133
AWV 200	0.746031746	0.380859893	0.262290616	0.266223919	0.668879135
MDS 200	0.732664996	0.371307214	0.226238672	0.267356477	0.607514701

Table 3.21: Results for Foursquare DT1.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.887218045	0.398256756	0.320725011	0.395149528	0.401413239
AWV 50	0.904761905	0.505069034	0.438593015	0.518028604	0.49274206
MDS 50	0.89390142	0.39499897	0.323143947	0.395981215	0.394021585
PCA 100	0.873015873	0.376756341	0.289354017	0.400611698	0.355582361
AWV 100	0.884711779	0.35844367	0.268995105	0.400944522	0.324089582
MDS 100	0.854636591	0.377612135	0.283423014	0.321749605	0.456947953
PCA 200	0.874686717	0.326303504	0.247282108	0.342929273	0.311215282
AWV 200	0.886382623	0.395980062	0.303883209	0.44741453	0.355152029
MDS 200	0.893065998	0.461813202	0.392690903	0.471080118	0.452903843

Table 3.22: Results for Foursquare DTN.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.810559006	0.358148107	0.234048804	0.299351526	0.44568688
AWV 50	0.836438923	0.451222703	0.336497634	0.374448235	0.567599713
MDS 50	0.784679089	0.392395666	0.24443375	0.313753839	0.523646773
PCA 100	0.824016563	0.361529188	0.239800468	0.316458032	0.421570923
AWV 100	0.867494824	0.513755413	0.413110212	0.434108709	0.629194937
MDS 100	0.713250518	0.390532004	0.227207011	0.280588887	0.642142724
PCA 200	0.8436853	0.40092042	0.291657581	0.35733735	0.456611589
AWV 200	0.865424431	0.513988009	0.41302185	0.445389498	0.60756459
MDS 200	0.637681159	0.39650034	0.213717085	0.262253502	0.812329599

Table 3.23: Results for Geonames SVM.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.680124224	0.29478471	0.122501925	0.201991012	0.545286876
AWV 50	0.713250518	0.25377	0.100391576	0.183142726	0.413064417
MDS 50	0.784679089	0.261891846	0.136012091	0.2084956	0.352053677
PCA 100	0.841614907	0.237260066	0.130180867	0.284786642	0.203327746
AWV 100	0.755693582	0.365164916	0.192834906	0.305374281	0.454069216
MDS 100	0.607660455	0.227454345	0.032886481	0.167615777	0.353738392
PCA 200	0.680124224	0.189278463	0.003823489	0.133089599	0.327578072
AWV 200	0.841614907	0.367019639	0.249986812	0.373118265	0.361117172
MDS 200	0.796066253	0.271625693	0.134331393	0.24476518	0.305108249

Table 3.24: Results for Geonames DT3.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.69047619	0.305148854	0.130369825	0.21577343	0.520918146
AWV 50	0.738095238	0.310954829	0.15490415	0.238367125	0.447108622
MDS 50	0.694616977	0.33974778	0.159514116	0.2446167	0.555959698
PCA 100	0.67805383	0.237734434	0.065010282	0.164848865	0.426151008
AWV 100	0.754658385	0.323247726	0.176195969	0.247133088	0.467114564
MDS 100	0.531055901	0.24795975	0.02750648	0.163433649	0.513575287
PCA 200	0.6563147	0.259807247	0.073980668	0.178667481	0.475958102
AWV 200	0.85300207	0.173697283	0.090449158	0.20356175	0.151474514
MDS 200	0.667701863	0.229791347	0.036946057	0.151458025	0.475950563

Table 3.25: Results for Geonames DT2.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.502070393	0.301104799	0.061819352	0.198359173	0.624669842
AWV 50	0.660455487	0.313961336	0.134392586	0.224578492	0.521532785
MDS 50	0.626293996	0.349125673	0.147011085	0.243137775	0.61892611
PCA 100	0.594202899	0.291992239	0.100440755	0.191654145	0.612833373
AWV 100	0.657349896	0.325658374	0.14728302	0.235431725	0.528013996
MDS 100	0.553830228	0.297616636	0.058027099	0.206694373	0.531351498
PCA 200	0.602484472	0.285530102	0.095389029	0.190597186	0.568877329
AWV 200	0.593167702	0.308568052	0.105417935	0.203284664	0.640064657
MDS 200	0.685300207	0.272314442	0.093166554	0.218701726	0.360748654

Table 3.26: Results for Geonames DT1.

	avg_acc	avg_f1	avg_kappa	avg_prec	avg_recall
PCA 50	0.81884058	0.206287264	0.089323711	0.196516538	0.217080418
AWV 50	0.83126294	0.257954417	0.149132237	0.24187974	0.27631775
MDS 50	0.803312629	0.265083269	0.11955942	0.268740939	0.261523827
PCA 100	0.820910973	0.24285113	0.10287381	0.286105247	0.210957969
AWV 100	0.845755694	0.274948597	0.176276415	0.27827957	0.271696423
MDS 100	0.82815735	0.194417219	0.088608744	0.182779779	0.207637318
PCA 200	0.801242236	0.239213047	0.101080713	0.230680272	0.248401316
AWV 200	0.8126294	0.332311235	0.192696161	0.296749178	0.377557229
MDS 200	0.844720497	0.294533825	0.15434765	0.380248649	0.240353795

Table 3.27: Results for Geonames DTN.

Table 3.28: Results for Genres SVM.

Table 3.29: Results for Genres DT3.

Table 3.30: Results for Genres DT2.

Table 3.31: Results for Genres DT1.

Table 3.32: Results for Genres DTN.

Table 3.33: Results for Keywords SVM.

Table 3.34: Results for Keywords DT3.

Table 3.35: Results for Keywords DT2.

Table 3.36: Results for Keywords DT1.

Table 3.37: Results for Keywords DTN.

Table 3.38: Results for Ratings SVM.

Table 3.39: Results for Ratings DT3.

Table 3.40: Results for Ratings DT2.

Table 3.41: Results for Ratings DT1.

Table 3.42: Results for Ratings DTN.

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

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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 non-commercially. 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.