

Representation Learning for Words and Entities

by

Pushpendre Rastogi

A dissertation submitted to The Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland

March, 2019

© 2019 by Pushpendre Rastogi

All rights reserved

Abstract

This thesis presents new methods for unsupervised learning of distributed representations of words and entities from text and knowledge bases. The first algorithm presented in the thesis is a multi-view algorithm for learning representations of words called Multiview Latent Semantic Analysis (MVLSA). By incorporating up to 46 different types of co-occurrence statistics for the same vocabulary of english words, I show that MVLSA outperforms other state-of-the-art word embedding models. Next, I focus on learning entity representations for search and recommendation and present the second method of this thesis, Neural Variational Set Expansion (NVSE). NVSE is also an unsupervised learning method, but it is based on the Variational Autoencoder framework. Evaluations with human annotators show that NVSE can facilitate better search and recommendation of information gathered from noisy, automatic annotation of unstructured natural language corpora. Finally, I move from unstructured data and focus on structured knowledge graphs. I present novel approaches for learning embeddings of vertices and edges in a knowledge graph that obey logical constraints.

Keywords: Machine Learning, Natural Language Processing, Representation Learning, Knowledge Graphs, Entities, Word Embeddings, Entity Embeddings.

Thesis Committee

Primary Readers

Benjamin Van Durme (Primary Advisor)
Assistant Professor
Department of Computer Science
Johns Hopkins University

Raman Arora
Assistant Professor
Department of Computer Science
Johns Hopkins University

Mark Dredze
Associate Professor
Department of Computer Science
Johns Hopkins University

Acknowledgments

First, I will like to thank my dissertation committee, Benjamin Van Durme, Raman Arora, Mark for all of their time and help. Without their advice, I will not have been able to finish this thesis. I am very grateful to the following people:

Benjamin Van Durme, my advisor: Ben admitted me to Hopkins and mentored me for six years. He worked tirelessly to improve my shortcomings, gave copious advice to me on writing, coding, and many other aspects of graduate life and did not give up even when I was slow on the uptake. Ben is an excellent thesis advisor, and he made me realize that I need to look at the bigger picture and stay focused on a topic to achieve results.

Raman Arora, Thank you from the bottom of my heart for your advice and support. I thoroughly enjoyed every time I worked with you, and you taught me a lot about machine learning. My most significant accomplishments resulted from my collaborations with you, and for that, I am genuinely grateful.

Jason Eisner, your knowledge, diligence, and technical prowess are truly admirable, and I learned a lot from working with you. You worked with me for one year and taught me a lot in the process. My collaboration with Jason resulted in an excellent paper which unfortunately I could not include in this

thesis because it was on a different topic.

Professors James Spall, Vince Lyzinski and Amitabha Basu from the AMS department for helping me find answers to many research problems that I got stuck on. Professor Spall taught me about stochastic optimization, and I also collaborated with him on this area which resulted in a CISS publication. Vince introduced me to the stochastic block models, and the problem of Vertex Nomination and a significant portion of this thesis was work done in collaboration with him. Although I never got the chance to collaborate and publish with Amitabha formally, I brainstormed with him about logically constrained embeddings with him, and he was always charitable with his time incredibly insightful and helpful.

I am also grateful to Professors Kevin Duh and Aaron Steven White, with whom I collaborated on the recasting semantics paper for all their hard work and tenacity in seeing the project to the end.

I will also like to thank Ruth Scally and Carl Pupa who have been incredibly helpful to not just me, but I think the whole CLSP department. Ruth went beyond her duty and helped me sort out some CPT-related issues in the summer of 2017, and I am ever grateful to her for that.

I will also like to thank all of the student collaborators that I have had the pleasure of working with. Ellie Pavlick and Juri Ganitkevich whom I assisted on the PPDB project. My collaboration with Ellie and Juri helped me get started on my research journey and resulted in my most cited paper to date. Manaal Faruqui – Ph.D. from CMU – who read my paper on MVLSA and invited me to collaborate with him to explicate problems of word embedding evaluation. That paper is now the second most cited paper on my resume! Ryan Cotterell, who is now a rising faculty member at Cambridge and is an

inspirational figure in my mind, for his insight and initiative that made my WFST paper with Jason possible. He mentioned the importance of tying the weights required for copying characters in previous literature. I realized that this trick could be implemented even in the neural architecture that we had and coded it and immediately our experimental results improved beyond SOTA. He is a smart researcher, and I am grateful to have worked with him. I also enjoyed my collaboration with Jingyi Zhu who worked with me on the Efficient Adaptive SPSA paper. Frank Ferraro, and Travis Wolfe, both one year senior to my Ph.D. cohort, with whom I published a paper each. They taught me about the tricks of Bayesian inference, max-margin learning and factor graphs. Rachel Rudinger who was my colleague and also a great friend who always gave well-reasoned and rational advice. Finally, I will like to thank Adam Poliak who has been both a great friend and a great research collaborator. We have published four papers together so far, and I will gladly work with him again given another chance. Finally, I will like to mention collaborators with whom I worked, but I was not to publish anything with just because I did not work smart enough. Nicholas Andrews, Mo Yu, Dingquan Wang, Nanyun Peng, and Elan Hourticolon-Retzler, hopefully, I will be to able to make up for the missed opportunities in future.

The Johns Hopkins University is a great university because of all the great students who come here. I enjoyed not only my collaborations with fellow students, but I also cherished their friendship. Finally, I will like to thank Adi Renduchintala, Gaurav Kumar, Keith Levin, Tongfei Chen, Anirbit Mukherjee, Skyler Kim, Tim Vieira, Keisuke Sakaguchi, Ting Hua, Corbin Rosset, Poorya Mianjyi, Inayat Ullah, Yasamin Nazari, Xuchen Yao, Patrick Xia, Seth Ebner, Ryan Culkin, Elias Stengel-Eskin and Andrew Blair-Stanek, for their friendship.

Table of Contents

Table of Contents	viii
List of Tables	xiii
List of Figures	xv
1 Introduction	1
1.1 Thesis Outline	4
1.2 Thesis Statement	5
2 Background and Motivation	6
2.1 Unsupervised Representation Learning	6
2.1.1 Shallow Representation Learning	7
2.1.1.0.1 Probabilistic PCA	8
2.1.2 Deep Representation Learning	10
2.1.2.1 Variational Autoencoders	11
2.1.2.1.1 Relation to Expectation Maximization	14
2.1.3 Multiview Representation Learning	15
2.1.3.1 Canonical Correlation Analysis	18

2.1.3.1.1	Computational Aspect: Algorithms for CCA	19
2.1.3.1.2	Nonlinear CCA	19
2.1.3.2	Generalized CCA	20
2.2	Information Retrieval and Set Completion	21
2.2.1	Set Completion and Variants	23
2.2.1.1	Examples of Set Completion Tasks	23
2.2.1.2	Methods for Set Completion	25
2.2.2	Existing Work on Entity Search	28
2.2.3	Distributed Representations of Knowledge Graphs	30
3	Multiview LSA	31
3.1	Introduction	32
3.2	Motivation	33
3.3	Proposed Method: MVLSA	34
3.3.1	Computing SVD of mean centered X_j	37
3.3.2	Handling missing rows across views	38
3.4	Data	40
3.4.1	Significance of comparison	43
3.5	Experiments and Results	45
3.6	Previous Work	51
3.7	Conclusion	53
4	Neural Variational Set Expansion	56
4.1	Introduction	57
4.2	Related Work	60

4.3	Notation	63
4.4	Baseline Methods	64
4.4.1	BM25	64
4.4.2	Bayesian Sets	65
4.4.3	Word2Vecf	66
4.4.4	SetExpan	66
4.5	Neural Variational Set Expansion	67
4.5.1	Inference Step 1: Concept Discovery	67
4.5.2	Inference Step 2: Entity Ranking	71
4.5.3	Unsupervised Training	72
4.5.4	Support for weighted queries	73
4.6	Interpretability	74
4.6.1	Query Rationale	74
4.6.2	Result Justifications	76
4.6.3	Weighted queries	76
4.7	Comparative Experiments	77
4.7.1	Dataset	78
4.7.2	Implementation Details	79
4.7.3	Experimental Design	80
4.7.4	Results	82
4.8	Analyzing Interpretability	84
4.8.1	Understanding the concept space	85
4.8.2	Weights & Query Rationale	86
4.9	Conclusion	87

5	Knowledge Base Embeddings under Logical Constraints	89
5.1	Introduction	90
5.2	Related Work	93
5.2.1	Methods for Knowledge Base Completion	93
5.2.1.1	Logically Constrained Representations for KBC	95
5.3	Theoretical Analysis of RESCAL	97
5.3.1	Proof of Theorem 1	99
5.3.2	Experimental Results	101
5.3.2.1	Experiments On Simulated Data	101
5.3.2.2	Experiments On WordNet	103
5.3.3	Discussion	106
5.4	Training Relation Embeddings under Logical Constraints	107
5.4.1	Constraints for Logical Consistency: Relational Implication	111
5.4.1.1	Reverse Relational Implication and Symmetry .	113
5.4.1.2	Entailment	113
5.4.1.3	Property Transitivity	114
5.4.1.4	Type Implication	116
5.4.2	Evaluating Logical Deduction and KBC on WordNet . .	118
5.4.3	Evaluating Link Prediction on WordNet	120
5.4.4	Discussion	122
5.5	Conclusion	123
6	Comparative Experiments	124
6.0.1	Hypothesis	126
6.1	Coreferent Mention Retrieval	126

6.1.1	Experiments	128
6.1.1.1	Mention Featurization	129
6.1.1.2	Learning Entity Embeddings	130
6.1.2	Results and Discussion	130
6.1.2.1	Performance Comparison between MVLSA and VAE	130
6.1.2.2	The influence of Decoder Type on VAE	132
6.1.2.3	The effectiveness of VAE training	132
6.2	Named Entity Disambiguation	132
6.2.1	Entity Embedding: Methods and Data	134
6.2.1.1	Data Sources	134
6.2.1.2	Methods	135
6.2.1.2.1	Max-Margin Entity Embedding	136
6.2.1.2.2	Variational Entity Embedding	137
6.2.1.2.3	Null Objective Entity Embedding	138
6.2.2	Related Works	139
6.2.3	Experiments and Results	139
6.2.3.1	Entity Relatedness:	141
6.3	Conclusion	142
7	Concluding Remarks	144
A	Appendix	146
A.1	Bayesian Sets	146
A.1.1	Binarizing feature counts	148
A.2	Ranking methods	148

List of Tables

3.1	Dataset statistics for MVLSA experiments	42
3.2	Testsets for MVLSA evaluation.	42
3.3	MVLSA performance versus non-linear pre-processing.	46
3.4	Performance versus truncation threshold hyperparameter.	46
3.5	Performance versus intermediate dimensionality.	47
3.6	Performance versus MVLSA embedding dimensionality.	48
3.7	Performance versus the support threshold.	49
3.8	Ablation tests for MVLSA	50
3.9	Comparison of MVLSA to Glove and Word2Vec.	51
4.1	Examples of queries for NVSE evaluation.	82
4.2	NVSE evaluation results (abridged)	82
4.3	NVSE evaluation results (Longform)	83
4.4	Concepts corresponding to NVSE embedding components.	85
4.5	Example of Query Rationales output from NVSE	85
5.1	RESICAL accuracy in different training scenarios.	104
5.2	Accuracy on WordNet for Logically constrained embeddings.	105
5.3	Some examples of errors by RESICAL	107

5.4	Instantiations of the BPR objective.	110
5.5	Sufficient constraints for enforcing implication.	115
5.6	Sufficient constraints for entailment.	117
5.7	Sufficient constraints for property-transitivity.	117
5.8	Example of a logical deduction problem	118
5.9	Results for the ELKB method	119
5.10	Comparative Results for ELKB models	121
5.11	Error analysis of logically constrained model	122
6.1	Contextual mention retrieval results	131
6.2	Micro F1 results for the same NED model but with entity embeddings pre-trained with different objectives. ✓ and ✗ in the second column indicates whether the learning algorithm had access to the mention context or not, respectively. The top two numbers in the dataset columns indicate the test-set size and the recall of the top-30 candidate entities.	141
6.3	Entity Relatedness Evaluation Results.	142

List of Figures

2.1	Pedagogical example to motivate unsupervised learning.	10
2.2	Graphical depiction of EM optimization	15
3.1	Input datasets for MVLSA.	41
4.1	Typical NLP pipeline for Named Entity Recognition and Linking.	59
4.2	The NVSE generation and inference models.	68
4.3	Example Mturk HIT for NVSE evaluation.	81
5.1	Illustration of Holographic Constraints.	94
5.2	A toy KB with one transitive relation.	100
5.3	Illustration of \mathcal{E} and \mathcal{E}^{rev}	102

Chapter 1

Introduction

People communicate with each other about entities in the real world using natural language. Due to the increasing digitization of communication and advancement of the internet, large natural language corpora are now available for computational analysis. Beneath the sizeable natural language corpora composed of words and sentences, lie pools of information about real-world entities, such as the names and affiliations of people, and details about states and nations. My goal for this thesis is to develop methods for learning distributed representation of words, and entities, that can improve Natural Language Processing (NLP) systems, and facilitate better search and presentation of information inside unstructured natural language data.

Words are a fundamental unit of natural language. Automatically learning about words, and quantifying this information, can ultimately help many NLP tasks. One of the earliest methods for learning dense representations of words was the linguistic vector space model called Latent Semantic Analysis (LSA) (Landauer and Dumais [1997](#)). LSA has been successfully used for information retrieval, but it has a limitation because it uses only a single view of

the data via a single word co-occurrence matrix. My **first contribution** in this thesis is an algorithm called **MultiView LSA** (MVLSA) (Rastogi, Van Durme, and Arora 2015). MVLSA overcomes the limitation of LSA to a single view because it can use an arbitrary number of views of data.

Generally speaking MVLSA is a multi-view algorithm and multiple view of data can help learning algorithms in two principle ways. First, the access to multiple views can help a learning algorithm to extract useful features that generalize across tasks and suppress spurious correlations that are dominant in one view and missing from the other. Second, multiple views may bring together complementary sources of information which a learning algorithm can combine to better learn the similarity between entities. More specifically, MVLSA is invariant to linear transformations of the data and this can also be considered as an advantage of MVLSA in comparison to LSA. I show that using a large number of views containing diverse sources of information improves the quality of the MVLSA word representations on many NLP tasks and makes them competitive with other popular word representation learning methods.

On the surface level, natural language is composed of words and sentences, but at a deeper, more conceptual, level these words and sentences convey information about real-world entities. Therefore, I claim that learning about entities can be even more useful than learning word representations in some fields of application. Consider the field of Information Retrieval for example. Information Retrieval (IR) is concerned with the search and presentation of information inside semi-structured and unstructured sources of data. Even though *Keyword based IR*, in which a user inputs a query in the form of a list of keywords, is the standard way of interacting with industrial IR systems

such as GOOGLE and BING, the field of IR is not restricted to keyword-based retrieval of documents. In the IR community, Entity Set Expansion (ESE)¹ is an established task of recommending entities² in a knowledge graph that are similar to a provided seed set of entities. Even small improvements in this task can have a tremendous impact on many fields.

For instance, imagine a physician trying to pinpoint a specific diagnosis or a security analyst investigating a terrorist network. In both scenarios, a *domain expert* may try to find answers based on prior known, relevant entities – such as a list of diagnoses with similar symptoms that a patient is experiencing or, a list of known terrorists. Instead of manually looking for connections between the known entities, *searchers* can save time by using an automatic *Recommender* that can recommend relevant entities to them. My **second contribution** in this thesis is the **Neural Variational Set Expansion** (NVSE) algorithm (Rastogi et al. 2018) that can operate on noisy knowledge graphs constructed automatically from a natural text document and recommend relevant entities. NVSE learns a probabilistic representation of an arbitrary subset of knowledge graph entities and uses this representation for the task of Entity Set Expansion. Through extensive experiments against existing state-of-the-art methods, I show that the NVSE algorithm can outperform existing methods for Entity Set Expansion.

Although one can learn much about entities in the world by the computational analysis of words associated to them – indeed, the NVSE method was based on this intuition – but there are important cases where the information about entities is stored in the form of a knowledge graph. A **Knowledge Graph** (KG)

¹ESE is also called Entity Recommendation in literature.

²Entities are also called ITEMS or ELEMENTS in the literature.

is a collection of relations of inter-connected entities. Large-scale semi-manually constructed KGs such as Freebase and YAGO2 have been heavily used for NLP tasks such as Relation Extraction, Question Answering, and Entity Recognition in Informal Domains. In the final part of this thesis, I consider the task of Knowledge Base Completion and propose methods for learning representations of knowledge graph entities that are not derived from the text. My **third contribution** is a method for learning embeddings of entities in Knowledge Bases that obey logical constraints (Rastogi, Poliak, and Van Durme 2017). I show that the proposed algorithm performs better than other baseline systems.

1.1 Thesis Outline

Chapter 3 presents the MVLSA algorithm for learning word embeddings from multiple sources of data and compares its performance to other state-of-the-art methods. Through experiments on a large number of word-similarity and word-analogy tasks, I show that the MVLSA embeddings are competitive with other methods for learning word embeddings.

Chapter 4 describes the NVSE algorithm for recommending entities grounded in natural language text. This task is called Entity Set Expansion (ESE). The NVSE algorithm is based on the Variational-Autoencoder (VAE) framework for training deep-generative models. Through human evaluations conducted on the Mechanical Turk platform, we verified that the NVSE algorithm outperforms pre-existing state of the art ESE methods.

Chapter 5 presents the logically constrained representation learning algorithm and compares it to other methods for learning representations of KB

entities.

Chapter 6 This chapter compares the various word-level and entity level algorithms developed in the previous chapters with each other on a few benchmark tasks.

Chapter 7 This chapter summarizes the contributions of the thesis and outlines directions for future work.

1.2 Thesis Statement

In this thesis, I present new algorithms for learning representation of words and entities from multiple views of data. I show that the proposed MVLSA algorithm is a generalization of the classical LSA method to multiple views of data and that incorporating various co-occurrence matrices for learning word representations improves the quality of the learned word representations. I then present a deep generative model for learning representations of entities present in natural language text and I also present the results of an approach I developed for enforcing logical constraints on the representations learned for representing entities in a knowledge base. Finally, I compare the algorithms developed in the thesis on the benchmark tasks of Contextual Mention Retrieval and Entity Disambiguation.

Chapter 2

Background and Motivation

This chapter provides the background and terminology necessary for understanding the methods used throughout this thesis. Since later chapters will refer back to these sections, one may choose to skim this chapter and refer back to it as a reference.

2.1 Unsupervised Representation Learning

At a high-level machine learning algorithms can be divided into two classes based on the available data and the type of task that is being performed, Supervised Learning and Unsupervised Learning. Supervised machine learning methods receive a *labeled* dataset containing pairs of inputs and outputs, and the goal is to construct a decision rule, that has high accuracy, based on the labeled data. On the other hand, unsupervised learning receives only a large dataset of input data, and the goal is to learn the regularities and patterns in the input data. The learned representations can be evaluated either by using the learned representations in a downstream task or by evaluating intrinsic properties of the representations such as invariances and nearest neighbors in the space of the

learned representations.

Unsupervised learning can be highly beneficial, in comparison to supervised learning, because of the absence of a single task and lack of sufficient labeled data. In such scenarios, learning the similarity between instances through unsupervised learning and leveraging that information can help to significantly reduce the sample complexity of learning. There are numerous examples of such applications. For example, (Nigam et al. 1998) used the expectation maximization algorithm over unlabeled data to learn feature weights for a Naive-Bayes classifier and showed that the number of samples required to achieve the same accuracy as a fully supervised naive bayes classifier decreased by as much as 50%.

Besides the obvious benefit of reducing the requirement on the number of labeled samples, unsupervised learning can also help by making the learnt features more task-agnostic. This is because, in contrast to supervised learning, unsupervised learning does not fit its features to the labels provided for a single task. Therefore the representations learnt during unsupervised learning can help in scenarios such as multi-task learning (Liu et al. 2016b; He and Lawrence 2011) and few shot learning (Fu et al. 2015).

2.1.1 Shallow Representation Learning

Principal Component Analysis (PCA) is one of the earliest statistical methods for unsupervised representation learning. Commonly PCA is known to be just an algorithm for linear dimensionality reduction shown in Algorithm 1. However, PCA models the data with a single latent subspace, and it can be interpreted not just as a procedure for linear dimensionality reduction but also as a *shallow* unsupervised representation learning algorithm, because the singular vectors

Algorithm 1 The PCA Algorithm

- 1: **Given:** We are given N data points each of which is d dimensional.
Let the i^{th} observation be called x_i and, let X be a $d \times N$ real matrix whose i^{th} column contains x_i .
Finally, let $\mathbf{1}$ be a column vector with N rows whose elements are all 1.
 - 2: **function** PCA(X, k)
 - 3: Let $\mathbf{x} = \frac{1}{N} \mathbf{1}^T X$ \triangleright \mathbf{x} is simply the sample average, i.e. $\mathbf{x} = \frac{1}{N} \sum_{i=1}^N x_i$.
 - 4: Let $\Sigma^{(k)}, U^{(k)}$ equal the top k singular values and corresponding left singular vectors of $X - \mathbf{1}\mathbf{x}$.
 - 5: **return** $\mathbf{x}, U^{(k)}, \Sigma^{(k)}$
 - 6: **end function**
-

$U^{(k)}$ constitute an orthogonal basis for the optimal k dimensional linear subspace that *best* encodes $X - \mathbf{1}\mathbf{x}$ according to the Frobenius norm of the total training error matrix. This interpretation of PCA as the solution of the optimal subspace learning problem is also referred to as the Geometric View or Synthesis View of PCA in the literature.

2.1.1.0.1 Probabilistic PCA Although, the Geometric View of PCA shows us that the projection matrix output by PCA maps a datapoint to a subspace which is *closest* to the training data, and therefore it motivates PCA as a representation learning algorithm. However, there is a more modern, probabilistic view of PCA, which frames PCA as a latent variable model and gives excellent insight into the type of representations that PCA learns. The probabilistic view of PCA was simultaneously introduced by (Tipping and Bishop 1999) and (Roweis 1998) where they showed that the output of the PCA algorithm could be used to estimate the parameters of a particular type of directed graphical model. Specifically, they considered the problem of parameter estimation for the following model.

Assume that the i^{th} observation is generated as follows, first a latent variable $z_i \in \mathbb{R}^k$ is sampled from a normal distribution $\mathcal{N}(\mathbf{0}, I)$. Then conditioned on the value of z_i , the vector $x_i \in \mathbb{R}^D$ is drawn from $\mathcal{N}(Mz_i + \nu, \sigma^2 I)$. Here M is the transformation matrix, ν is a vector and σ is a positive scalar. In other words

$$p(x_i|z_i, M, \nu, \sigma^2) = \mathcal{N}(Mz_i + \nu, \sigma^2 I). \quad (2.1)$$

According to the above model the observed data is drawn from a continuous mixture model and since this is an unsupervised probabilistic model it makes sense to talk about the MLE estimate of M , \hat{M} . (Tipping and Bishop 1999) showed that \hat{M} can be computed as:

$$\hat{M} = U^{(k)} \left(\frac{\Sigma^{(k)}}{N} - \sigma^2 I \right)^{1/2} R,$$

where R is an arbitrary orthonormal matrix, $U^{(k)}$ is an orthogonal matrix containing the top k eigen vectors, and $\Sigma^{(k)}$ is a diagonal matrix containing the top k eigen-values, of the empirical covariance matrix $1/n X^T X$. Therefore, if $\sigma = 0$ then $U^{(k)} \left(\frac{\Sigma^{(k)}}{N} \right)^{1/2}$ is the maximum likelihood estimator of M and $U^{(k)} \left(\frac{\Sigma^{(k)}}{N} \right)^{1/2} x$ is the *plugin estimate* of z . This interpretation also clarifies the relation between PCA and other probabilistic methods such as factor analysis (Spearman 1904; Thurstone 1947). For example we can easily see from Eq. 2.1 that probabilistic PCA is just a more restricted form of factor analysis where the error variance along each dimension is assumed to be the same. In contrast to Eq. 2.1 factor analysis () assumes that the errors along each dimension can have different variance but are still uncorrelated, i.e.

$$p(x_i|z_i, M, \nu, \sigma^2) = \mathcal{N}(Mz_i + \nu, \begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots \\ 0 & \sigma_2^2 & 0 & \dots \\ 0 & 0 & \sigma_3^2 & \dots \\ 0 & \dots & 0 & \sigma_d^2 \end{bmatrix}). \quad (2.2)$$

2.1.2 Deep Representation Learning

Pedagogically simple examples of the utility of unsupervised learning arise in learning disentangled representations of low dimensional manifolds such as the swiss-roll dataset shown in Figure 2.1. Figure 2.1 shows that learning a

The Swiss Roll Dataset.

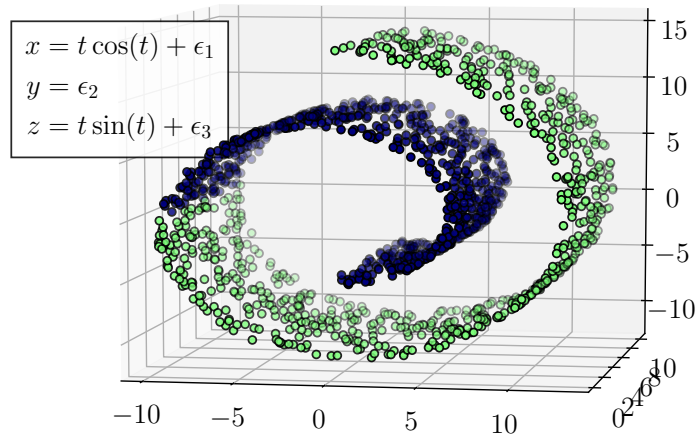


Figure 2.1: The Swiss-Roll Dataset, where 3 features x, y, z are observed. x, z depend on a single underlying parameter t , and y is just noise. Learning a map from x, y, z to t from unsupervised data can decrease the number of samples needed to discriminate between green and blue points accurately.

good representation of the observed data can be fruitful later on for supervised learning. Locally Linear Embeddings (LLE) (Saul and Roweis 2003), Laplacian Eigenmaps (Belkin and Niyogi 2003) and IsoMaps (Tenenbaum, De Silva, and Langford 2000) are a few of the best-known unsupervised machine learning algorithms that work on this principle. A seminal paper by (Bengio et al. 2004) unified these three algorithms – amongst several others – by recasting the problem of unsupervised learning of embeddings to learning eigenfunctions of a

data-dependent kernel.

After the success of these early algorithms, the field of unsupervised representation learning grew rapidly. A significant development was the construction of algorithms such as the Restricted Boltzmann Machine (RBM) (Hinton and Salakhutdinov 2006) and Deep Belief Networks (Hinton, Osindero, and Teh 2006) for training deep neural networks from unsupervised data. These methods learned the parameters of neural networks which can convert sparse input data to dense, distributed, vector representations. Moreover, these representations were proved to be useful for real-world tasks such as Collaborative Filtering (Salakhutdinov, Mnih, and Hinton 2007).

More recently, a new method called the Variational Autoencoder (VAE) (Kingma and Welling 2014a; Rezende, Mohamed, and Wierstra 2014) for learning deep, non-linear, generative models from unlabeled data was proposed, which has resulted in tremendous advancements in unsupervised and semi-supervised machine learning (Kingma et al. 2014; Miao, Yu, and Blunsom 2016). In the following section, I give an overview of the VAE framework.

2.1.2.1 Variational Autoencoders

Generative modeling of data is a broad topic in data science. A generative model of a dataset can make its underlying factors of variations more explicit, and it can help us summarize and understand large amounts of data quickly. An example of a generative model of data is the Latent Dirichlet Allocation Topic Model (Blei, Ng, and Jordan 2003). A topic model can summarize a large dataset by discovering clusters of commonly co-occurring features. Many “component analysis” methods such as PCA and CCA discussed in previous

sections can be interpreted as statistical generative models as well (Tipping and Bishop 1999; Bach and Jordan 2005).

Latent variable models are a useful sub-type of generative model that can be useful in situations where the observed data lies in a high-dimensional space, but the elements of the data contain strong inter-dependencies. In common parlance, such data is said to lie on a low-dimensional manifold. If the inter-dependencies between the components can be de-coupled by introducing a small number of latent variables without introducing too much error, then such a latent variable model can be useful both for its predictive accuracy and also for its explanatory power. For example, suppose that we are trying to learn a generative model of human face images. The observed pixels in an image have to satisfy many constraints such as bi-lateral symmetry across the face, consistent skin coloring and relative proportion of eyes, nose, and ears. Because of all these constraints, the pixel intensity at the top-right corner of a face image may be highly correlated to the pixel intensity at the bottom-left corner in the general population of the entire dataset. However now consider a situation where we can stratify the dataset by the gender, age, race, and weight of the person. Within each stratum, the correlation between the top-left pixel intensity and the bottom-right pixel intensity will be closer to zero than the correlation in the general population. Even though these 4 variables are not observed in the dataset, by introducing these 4 factors as latent variables and then adding a conditional independence assumption amongst the observed variables given these unobserved variables we can make the model better suited to the data. Another motivation for latent variable models is that they are a mixture model and can approximate multi-modal distributions easily. Many excellent books on machine

learning – such as (Murphy 2012; Bishop 2006; MacKay 2002) – expand upon this point of view, and I will not expand upon this more.

The Variational Autoencoder is a framework for learning the parameters of a *latent variable* generative model from i.i.d. unlabeled data samples. Formally, say that we are given a dataset \mathcal{D} containing n i.i.d. samples of a random variable X . We posit that there exists a latent random variable Z such that instances of X are conditionally independent given Z . In other words we posit the following generative story:

$$Z \sim \pi(z), \quad X|Z \sim p_\theta(x|z) \quad (2.3)$$

Here θ parameterizes the conditional probability distribution of X given Z . According to this model the marginal distribution of X is given as

$$p(X) = \int_z p_\theta(x|z)\pi(z)dz \quad (2.4)$$

Maximum Likelihood Estimation (MLE) with regularization is perhaps the most common method for learning the parameters θ for a statistical model given \mathcal{D} . For simplicity I will omit discussion of the regularization for now and focus only on the likelihood function itself. The MLE procedure maximizes the likelihood of the parameters θ for a given dataset \mathcal{D} . Therefore the MLE procedure maximized the following objective:

$$\mathcal{J}_{\text{MLE}}(\theta) = \sum_{i=1}^n \log p_\theta(x_i) = \sum_{i=1}^n \log \int p_\theta(x_i|z)\pi(z)dz.$$

In some situations it may be possible to compute the above objective, for example in a discrete mixture model without priors on the mixture parameters but in general it is not possible to compute the above sum over the dataset. A common solution for such problems utilizes the following identity called the *Variational*

Identity which introduces a new distribution over the latent variables that I denote $q(Z)$

$$\log p(X) = E_{Z \sim q(Z)} \left[\log \frac{p(X, Z)}{q(Z)} \right] + \text{KL}(q(Z) \parallel p(Z \mid X)) \quad (2.5a)$$

$$\log p(X) = E_{Z \sim q(Z)} [\log p(X, Z)] + \mathbb{H}[q(Z)] + \text{KL}(q(Z) \parallel p(Z \mid X)) \quad (2.5b)$$

$$\log p(X) = E_{Z \sim q(Z)} [\log p(X|Z)] - \text{KL}[q(Z) \parallel \pi(Z)] + \text{KL}(q(Z) \parallel p(Z \mid X)) \quad (2.5c)$$

Identity (2.5a) can be easily verified simply by expanding the definition of the KL divergence. In the above identities, $q(Z)$ is actually shorthand for $q(Z \mid X, \phi)$ i.e., $q(Z)$ is an arbitrary distribution over the latent variables that can freely depend on the values of X and other parameters ϕ . The variational auto-encoder specifies a special type of $q(Z \mid X, \phi)$ which is parameterized as a differentiable neural network. I will give more details about specific architectures in Chapter 4.

2.1.2.1.1 Relation to Expectation Maximization Consider identity (2.5b) and note that if $q(Z)$ exactly equals $p(Z|X)$ then the $\text{KL}(q(Z) \parallel p(Z \mid X))$ term becomes zero. Moreover we get the formula that

$$p(X) = E_{Z \sim p(Z|X)} [\log p(X, Z)] + H[p(Z|X)] \quad (2.6)$$

Here the H operator computes the entropy of a distribution. The EM procedure discards the entropy of $p(Z|X)$ and optimizes the Joint Likelihood with Current Parameters to get the following iterative learning rule:

$$\theta_{t+1} = \arg \max_{\theta} E_{Z \sim p_{\theta_t}(Z|X)} [\log p_{\theta}(X, Z)],$$

So we can see that the Expectation Maximization procedure is simply a special case of the variational optimization procedure. However, this special case of

variational optimization enjoys a wonderful property of *Monotonicity*, i.e. the value of the objective $E_{Z \sim p_{\theta_t}(Z|X)} [\log p_{\theta}(X, Z)]$ increases with t . Figure 2.2 gives a graphical explanation of the EM procedure described above.

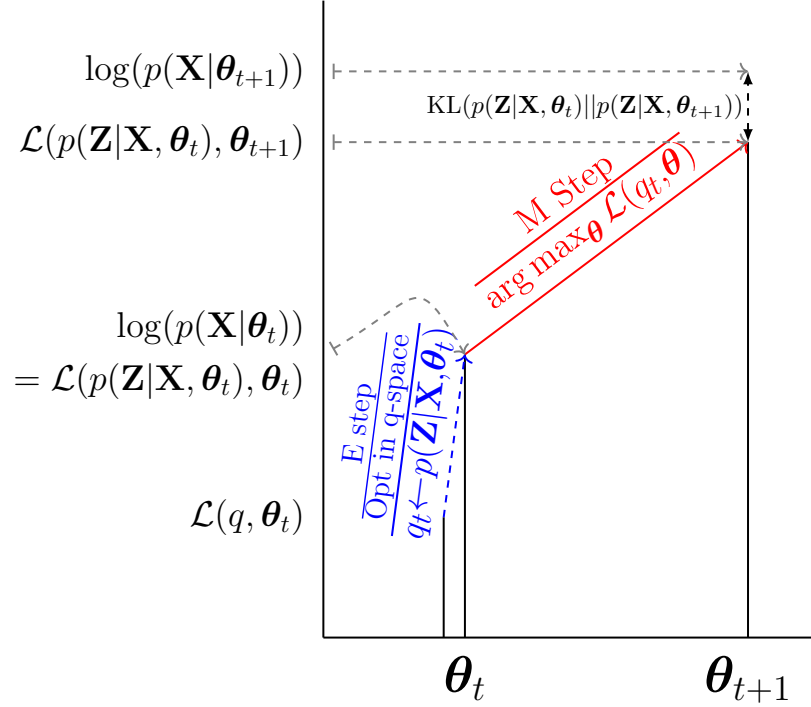


Figure 2.2: A graphical depiction of the EM iteration as coordinate ascent in the space of probabilities and parameters. Let $\mathcal{L}(q, \theta)$ denote $\int_z q(Z) \log \frac{p(X, Z|\theta)}{q(Z)}$. In the so-called E-step we optimize by setting $q^t = p(Z|X, \theta_t)$ and in the M-step we set θ by maximizing $\mathcal{L}(q_t, \theta_t)$.

2.1.3 Multiview Representation Learning

The goal of multi-view representation learning is to learn a single unified representation for data that is observed via multiple views/channels, but that has a single underlying source. There is no standard definition of a “view” in multiview learning and it can even overlap with multimodal learning (Ngiam et al. 2011) to some extent. Multiview learning can refer to any technique that learns one

classifier/regressor per view and a common underlying representation, that is unknown apriori. (Sridharan and Kakade 2008) state that the fundamental assumption underlying multi-view learning is that any of the views used for learning *alone* has sufficient information about the target.

There are two natural classes of multi-view learning algorithms that utilize this assumption. The first class comprises the algorithms such as Co-Training (Blum and Mitchell 1998) and co-regularization (Sindhwani, Niyogi, and Belkin 2005) which regularize the predictions of classifiers learnt from separate views of data to agree with each other. The second class of algorithms based on correlation analysis such as (Kakade and Foster 2007) and (Wang et al. 2015) focus on learning correlated representations of data using unsupervised learning methods and these *correlation analysis* based methods will be the focus of this thesis.¹

There are many natural applications for unsupervised multi-view learning especially in the field of natural language processing (NLP) because of the high dimensionality and sparsity of the bag-of-words feature representation that is typically employed for many NLP tasks. By properly utilizing multiple view of data we can learn better representations of data which can even provably reduce the number of samples required for learning. For example, consider models such as Glove (Pennington, Socher, and Manning 2014) and Word2Vec (Mikolov, Yih, and Zweig 2013) that learn a dense vector representation of a word from a single view of linguistic data such as a large corpus of natural language text sequences. However these methods are not able to distinguish between antonyms

¹We note that correlation analysis is not the sole method for unsupervised multi-view learning. A good example of such a technique is the work by (Ngiam et al. 2011) who presented an auto-encoder based framework for learning feature extractors that work well for feature learning from multimodal data.

such as the words GOOD and BAD because both the antonyms are used in very similar contexts. On the other hand by using an external view such as the dictionary entries for a word we can learn to distinguish between two antonyms. In this sense, the two views bring complementary information that a multi-view learning algorithms can utilize. Other successful applications include multiclass classification (Arora and Livescu 2014), clustering (Chaudhuri et al. 2009; Zhang et al. 2016) and ranking/retrieval (Vinokourov, Cristianini, and Shawe-Taylor 2003; Cao et al. 2018). Another application of multiview learning for natural language processing was presented by (Benton, Arora, and Dredze 2016) who also released a dataset for learning multiview representations of twitter users.²

There has also been work done on providing guarantees for performance improvement so that multiple views can be guaranteed to not hurt the performance of a regressor or classifier. A typical assumption that is utilized in Multiview learning in order to improve the sample complexity of a learning method is that the views of data are conditionally independent given the underlying common representation. Conventional methods for machine learning do not make this assumption, and they may concatenate all the features from the different views and treat the multi-view dataset as a single-view dataset. For example (Kakade and Foster 2007) showed that using unlabeled data from two views can reduce the sample complexity of prediction problems. Specifically they provided a semi-supervised algorithm which first uses unlabeled data to learn a kernel, and then regularized a ridge-regression classifier according to the learnt norm.

In the remaining part of this section, I describe a few classical multiview representation learning techniques that are pertinent to this thesis.

²https://www.cs.jhu.edu/~mdredze/datasets/multiview_embeddings

2.1.3.1 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA), first proposed by (Hotelling 1935), is a procedure that finds linear projections of two datasets such that the correlation between the two projections is maximized. Let X_1, X_2 be two centered and standardized matrices denoting two views of the data with n rows and d_1, d_2 columns respectively. Let $P_j = X_j U_j$ where $U_j \in \mathbb{R}^{d_j \times d}$ is a linear projection matrix and $j \in \{1, 2\}$. d_1, d_2 are the dimensionalities of X_1, X_2 respectively, and d is the dimensionality of the latent space. Let $\hat{\Sigma}_{jj'} = \frac{1}{n} X_j^T X_{j'}$. The CCA procedure determines projection matrices U_1, U_2 according to the following optimization problem

$$\begin{aligned} & \arg \max_{U_1, U_2} U_1 \hat{\Sigma}_{12} U_2 \\ & \text{subject to } U_1 \hat{\Sigma}_{11} U_1 = 1 \text{ and } U_2 \hat{\Sigma}_{22} U_2 = 1 \end{aligned} \quad (2.7)$$

(Hardoon, Szedmak, and Shawe-Taylor 2004) highlight two of the most important ways to motivate the CCA objective and to derive its solution and (Hastie, Buja, and Tibshirani 1995) highlight an interesting connections between CCA and Fisher's Linear Discriminant Analysis in the case of categorical classification, which I will not repeat here. Instead I list some of the important properties of the CCA procedure that are most relevant to us

- The CCA projections P_j are invariant to shifting and scaling the data.
- Let A, S, B denote the singular value decomposition of $\hat{\Sigma}_{11}^{-1/2} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-1/2}$. I.e.

$$\hat{\Sigma}_{11}^{-1/2} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-1/2} = ASB^T,$$

then $U_1 = \hat{\Sigma}_{11}^{-1/2} A$ and $U_2 = \hat{\Sigma}_{22}^{-1/2} B$.

2.1.3.1.1 Computational Aspect: Algorithms for CCA As described above the CCA learning problem can be reduced to solving a singular value decomposition problem of an asymmetric matrix $\hat{\Sigma}_{11}^{-1/2} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-1/2}$. However, empirically computing this matrix may be intractable because of the quadratically increasing memory requirement as the number of examples in the dataset increases. More recently a number of works have proposed more scalable methods for computing CCA such as (Ge et al. 2016; Arora et al. 2017; Gao et al. 2017; Allen-Zhu and Li 2017). For example, (Arora et al. 2017) proposed a convex relaxation of the original CCA optimization problem and they presented stochastic approximation algorithms for optimizing the resulting objective in a streaming setting. And they showed that their proposed stochastic approximation algorithm outperformed existing state-of-the-art methods for CCA on a real dataset.

2.1.3.1.2 Nonlinear CCA An interesting direction for generalizing Canonical Correlation is to use non-linear functions for projecting the views. Kernel CCA (AKAHO 2001; Hardoon, Szepesvári, and Shawe-Taylor 2004) and Deep Canonical Correlation Analysis (DCCA) by (Andrew et al. 2013) were two efforts in this direction. KCCA is a nonparametric method for learning non-linear transformations that produce high correlated projection in a reproducing kernel Hilbert space. On the other hand, DCCA trains two deep-neural networks to learn nonlinear transformations of respective data views by optimizing a regularized correlation objective.

2.1.3.2 Generalized CCA

Canonical Correlation Analysis is one of the earliest multiview learning algorithms; however, it is limited to only two views by construction. In order to remove this limitation, several generalizations of CCA have been proposed in the literature. (Kettenring 1971) proposed 5 possible ways of generalizing CCA, and all of those 5 methods possessed the special property that they reduced to standard CCA when using only two views of data. (Asendorf 2015) further extended the work by (Kettenring 1971) and proposed 20 possible generalization of CCA. Instead of reviewing all of the possible generalizations of the CCA objective I will focus on one particular variant of Generalized CCA – introduced by (Carroll 1968) – which Kettenring called the MAXVAR generalized CCA method. Like all the other variants studied by Kettenring MAXVAR GCCA projections are also equivalent to the standard CCA projections in the case that there are only two views of the data.

Let X_1, \dots, X_J be J observed views of the same underlying data, with $J \geq 2$. The MAXVAR GCCA procedure – which I will call GCCA from now on – finds J projection matrices $U_j \mid j \in \{1, \dots, J\}$, and a common latent representation G such that the sum of the squared correlations between the view-projections $X_j U_j$ and the latent representation G is maximized. Formally, the GCCA objective is:

$$\arg \max_{Y, U_1, \dots, U_J} \sum_{j=1}^J \text{trace}(Y^T (X_j U_j)) \text{ subject to } Y^T Y = I$$

This constrained maximization can be reframed as least squared error minimization optimization problem as follows

$$\arg \min_{Y, U_1, \dots, U_J} \sum_{j=1}^J \text{trace}((Y - X_j U_j)^T (Y - (X_j U_j))) \text{ subject to } Y^T Y = I \quad (2.8)$$

I will refer to this objective in Chapter 3.

2.2 Information Retrieval and Set Completion

The field of *Information Retrieval* (IR) is concerned with the search and presentation of information inside semi-structured sources. Examples of semi-structured information sources are web pages, images, research papers, and résumés. IR is different from querying databases because of the lack of precise semantics of data. For example, IR systems, such as Internet search engines, need to answer queries like, “What is the largest city in the world?” without asking the user, which attribute of a city should be used to sort the cities, or what is the precise definition of a “city”.³

Instead of asking for all sorts of clarifications IR systems work *intelligently* and they find documents that are most likely to contain the answer for a query. Therefore, IR systems are best thought of as fast and efficient statistical prediction engines which incorporate:

1. *A document level prior about the importance of a document.* For example, the Pagerank algorithm is an unsupervised method that utilizes the hyperlinks in web-pages to learn the prior probability of the importance of a web-page (Brin and Page 1998; Yin et al. 2016).⁴
2. *The probability of a document’s relevance to a query.* Search Engine Click-Logs that contain the URL that a person clicked amongst the search results

³Clearly not all users will be satisfied with the results, in which case they will modify the query and retrieve a new set of web pages.

⁴If the document collection does not contain hyperlinks, then other meta-data such as the length of the document, author information and last modification time can be used to model the importance of a document.

in response to a query provide a useful signal for estimating this. Finally, the document’s content can be analyzed to predict whether it is relevant to a question.

Even though *Keyword based IR*, in which a user inputs a query in the form of a few keywords is the standard method for interacting with well known IR systems such as Google and Bing, research in IR has not restricted to keyword-based retrieval, nor has it restricted to the retrieval of documents (Manning, Raghavan, and Schütze 2010). *Recommender Systems* on e-commerce websites that retrieve *items*, or *entities*, relevant to a customer, from a customer’s profile and a large corpus of customer-item interactions are examples of *non-keyword* IR systems that return items from a catalog without any textual query.⁵

Let us consider another application of non-keyword, non-document IR systems which will also motivate my research problem: Consider the situation of a recruiter who needs to find suitable candidates for a job from a large corpus of candidates. Moreover, the recruiter has access to the candidates’ friendship network, résumés, personal statements and publications. More specifically, the job may require people who possess a good understanding of “information retrieval” techniques and the “Hindi” language. However, it is possible that not every candidate’s profile contains that information. Instead, publishing in the SIGIR conference, or being a citizen of India, or being friends with multiple people like that may be good indicators of the above qualities. These correlated qualities could be inferred from examples of desirable entities. This example suggests that:

⁵Other examples of *non-keyword* IR systems are multimedia retrieval systems that allow users to use an audio recording or an image for retrieving the results. I will not focus on multimedia retrieval.

1. Graph structured side information about entities can be useful for IR.
2. Examples of relevant entities provide useful feedback to an IR system.
3. Entity retrieval is a useful task.

The problem of finding more items that are similar to a given set of items and the problem of finding items in response to a keyword query have both been studied extensively. In case the user does not provide any keyword query and only provides examples of items to be retrieved then the problem will be considered as the problem of *Set Completion*

2.2.1 Set Completion and Variants

Retrieving entities that are similar to a few example entities is an artificial intelligence task with broad utility.

2.2.1.1 Examples of Set Completion Tasks

The problem of finding a suitable candidate for a job can be framed as the problem of Set Completion if a few examples of suitable candidates are given and the system needs to rank the unknown candidates in a database according to their suitability for the job. This task of finding suitable candidates for a task is broadly referred to as the problem of *Expertise Retrieval* and it operationalized and evaluated in various ways (Balog 2012). Expert Retrieval is a special case of a more general problem called *Entity Retrieval* in which, a seed entity, a description of the relations between the target entity and the seed entity, and a few examples of the target entities are provided as inputs. Variants of this task,

depending on whether entities are provided as examples or not, have been run multiple times in the annual TREC conference (Balog 2012).

Another example of a set completion task is the *Document Routing* problem, in which documents related to a topic need to be retrieved given a natural language text describing a user’s information need and some example documents is another example of the set completion task. Document Routing was an important shared task evaluated during the early annual TREC conferences (Schütze, Hull, and Pedersen 1995).

A third example of the set completion problem is the task of *Item Recommendation* to customers from their purchase history and profile. If information about the purchase history of other customers is also available then the problem is called *Collaborative Filtering* otherwise the problem is called *Content Based Recommendation*. Note that technically the term *filtering* should be used if the task is to classify whether an entity lies in a set instead of ranking the remaining entities. Such a problem, where the set of example entities needs to be increased is also called the *Set Expansion* problem.

Set completion tasks where relational information amongst the entities is available were called the *Vertex Nomination* (VN) task by (Fishkind et al. 2015) and *Class-Instance acquisition* by (Talukdar and Pereira 2010). The Vertex Nomination terminology is apter in situations where the graphs are more homogeneous with lesser entity level features, and the edges between the entities are not too sparse.⁶ In the case where additional meta-data is available, and the graph is sparse, and there can be multiple possible types that a particular vertex

⁶Note that the research on VN is evolving and more algorithms are being introduced which get rid of some of these assumptions, and therefore these distinctions are not strict.

can take on then the terminology of *Class-Instance acquisition* is perhaps apter.

Finally, the task of *Query By Entity*, or *Query By Examples* has been studied in the database community and the semantic web community as a way of helping user’s interact with databases or knowledge graphs (Metzger, Schenkel, and Sydow 2013). In these tasks, a computer system needs to infer and execute an unknown query on the basis of a few examples of the results, and therefore this task too can be considered to be a set completion task.

2.2.1.2 Methods for Set Completion

One of the earliest methods for set completion was a patented approach called “Google Sets” (Tong and Dean 2008)⁷ that performed set completion by modeling the input examples as samples from a mixture of distributions over pre-existing lists. After receiving a few examples of entities, the mixture components were estimated and then new entities were generated from this distribution. Inspired by this approach (Ghahramani and Heller 2005) introduced the method of “Bayesian Sets”. The Bayesian Sets method ranks the entities by the ratio of two probabilities. The first probability measures whether the entity and the data were generated from the same parameters and the second probability measures whether the data and the entity were generated independently. Some theoretical results about the *stability* of the method in the presence of correlated features were presented in (Letham, Rudin, and Heller 2013). However, the method by itself does not provide guarantees about the *quality* of the rankings.

A different approach than the Bayesian sets method which creates a “profile” of the criterion for being in a set is to define a similarity function such that

⁷Note that although the patent application was filed in 2003 it was only granted in 2008.

new items that are most similar to the example entities get a high rank. SEAL and its variants (Wang and Cohen 2007, 2008) were early approaches that learned the similarity between the new entities and example entities using methods like “Random Walks”, and “Random Walks With Restart”. Other methods for computing similarity have also been explored such as the sum of cosine similarities amongst others. In modern parlance, any kernel method that computes similarities between two entities can be used for ranking the entities. The hyper-parameters for the kernel methods can be trained via 5-fold cross-validation on the training data.

A very different approach to this problem is to treat the problem as a binary classification task, where only positively labeled examples and unlabeled data is available. Viewed this way any generative probabilistic model usable for binary classification can be applied in a principled manner to this problem, for example, the generative Naive Bayes algorithm was applied for binary classification. (Nigam et al. 1998, 2000) used precisely this model with the EM algorithm to utilize unlabeled data to learn the parameters of a naive Bayes text classifier.

A different principled approach is to apply the PU learning framework (Denis 1998). Under the PU learning framework, the learning algorithm tries to minimize the total probability of labeling the unlabeled data as positive while holding the probability of correctly labeling the labeled data above the desired recall rate (Liu et al. 2002; Liu et al. 2016a). (Li et al. 2010b) compared the performance of a PU learning-based classifier to distributional similarity based methods and the Bayesian Sets method on the problem of entity set expansion. The distributional similarity based method ranks entities from the cosine similarity of its distributional signature (PMI features). They showed that their PU

learning method outperformed both distributional methods and the Bayesian Sets method. Recent work by (Natarajan 2015) and (Rao et al. 2015) amongst others applied PU learning methods to problems of matrix completion and collaborative filtering while incorporating additional graph information as well. Note that PU Learning is a very active research area and activity in this area is accelerating.

The problem of set completion can also be solved using the *learning to rank* approach (Li 2014). In the learning to rank framework, the set completion algorithm learns a pairwise decision function that receives two entities as inputs and decides which of those two entities is more likely to be a member of the set. This function can be trained so that it always picks the input labeled entities to be members of the set in comparison to unlabeled entities. Learning to rank methods are very popular in the Information Retrieval community (Manning, Raghavan, and Schütze 2008).

The problem of vertex nomination was tackled using the Adjacency Spectral Embedding method and its extensions by (Sussman et al. 2012; Fishkind et al. 2015) on communication graphs where the Stochastic Block Model is a reasonable approximation to the generative process for the observed graph. On the other hand, when the graphs were manually created knowledge graphs such as Freebase, or automatically extracted OpenIE knowledge graphs such as the Textrunner graph, which naturally exhibits sparsity, and bipartiteness, then this problem has been tackled using *Graph Based Semi-Supervised Learning* algorithms such as the Label Propagation algorithm, the Adsorption, and the Modified Adsorption algorithm by (Talukdar and Pereira 2010).⁸

⁸The framework of *querying by examples* on RDF triples provides a useful framework for

Finally, set completion can be reduced to the problem of *Link Prediction* or *Knowledge Base Completion* by adding a new meta-vertex that represents the set and then connecting the labeled entities in the training set to the meta-vertex. I can represent the likelihood of an unlabeled entity being part of the group as the score assigned to an edge that connects the meta-vertex to the unlabeled entity. See (Nickel et al. 2016) for a review of this area.

2.2.2 Existing Work on Entity Search

Research in entity search over large text corpora was accelerated with the start of the TREC entity retrieval and expertise retrieval tracks (Balog, Serdyukov, and Vries 2012; Balog 2012). These shared tasks considered the same problem as us, where a query was a bag of keywords, and the result of a query was a ranked list of individual entities, each of which was an answer. (Dalton, Dietz, and Allan 2014) further used knowledge graphs for feature expansion. One of the dominant methods was introduced by (Balog, Bron, and De Rijke 2011), which is based on entity language models and harnesses entity categories for ranking and for restricting answers to the desired type. However, these methods have been tested only on situations where large Wikipedia pages were available for estimating the language models associated with an entity. A similar approach, of creating entity language models, using only the text surrounding the mentions of an entity was earlier explored by (Raghavan, Allan, and McCallum 2004). However, they did not consider how to incorporate side information or relevance feedback.

framing the problem of set completion. This is also a large area, and I leave it out of this review.

Searching and exploring text corpora that are annotated with entities and linked to a KG has been addressed in various projects, most notably the Broccoli system (Bast et al. 2014), Facetedpedia (Li et al. 2010a), ERQ (Li, Li, and Yu 2012), STICS (Hoffart, Milchevski, and Weikum 2014), and DeepLife (Ernst et al. 2016). The work by (Agrawal et al. 2012) Expanded upon the work by (Li, Li, and Yu 2012) and added a notion of similarity between entities, which they called “near” queries. Additionally, they utilized the “Spreading Activation” method for including graph structure into the ranking of entities. Furthermore, they only considered the Wikipedia graph as an example.

(Sawant and Chakrabarti 2013) and (Joshi, Sawant, and Chakrabarti 2014) considered the problem of answering short keyword-based text queries over a combination of textual and structured data. Their approach was to jointly learn the segmentation, the entity, class and predicate interpretation of the input query (in text form), and the ranking of candidate results. They did not consider the problem of entity-based relevance feedback, however. (Yahya 2016) worked on supporting complex queries on knowledge bases that also contain textual web content in their fields. They called such knowledge graphs “Extended Knowledge Base”.

Recently (Savenkov and Agichtein 2016) and (Xu et al. 2016) considered the extended knowledge graph as a starting model for performing question answering over knowledge bases. Specifically, they showed that access to related text could improve the performance of various sub-components in information extraction and semantic parsing pipelines, such as entity linking and coreference resolution.

2.2.3 Distributed Representations of Knowledge Graphs

The singular vectors of word-document co-occurrence matrices were one of the first vector representations of words and documents used in the field of NLP and Information Retrieval. After the work of (Mikolov, Yih, and Zweig 2013; Mikolov et al. 2013) there was an explosion of activity in the area of learning vector representations of words, graphs, and other discrete structures. Interesting new directions were proposed by (Vilnis and McCallum 2015) and (Rudolph et al. 2016). (Vilnis and McCallum 2015) proposed to represent each word in a sequence by a Gaussian distribution, and this work was extended to learning representations for entities in a knowledge graph by (He et al. 2015). On the other hand, (Rudolph et al. 2016) proposed the EF-EMB model to represent the conditional distribution of a “related” entity given a “base” entity using exponential family distributions. They applied their model to the task of predicting a neuron’s activity from its neighbors’ activities.

Recently (He et al. 2015) applied the Gaussian Embedding method presented by (Vilnis and McCallum 2015) to learn vector representations of graph vertices, and they tested their learned representations on tasks such as link prediction.⁹

⁹The link prediction task aims to find the correct entity that should be linked to a given entity with a given relation and measures performance using IR metrics.

Chapter 3

Multiview LSA

The primary goal of this chapter¹ is to motivate and describe the Multiview LSA (MVLSA) algorithm which is a significant generalization of classical methods such as LSA. To that end, I compared the performance of MVLSA against single view LSA as well as other contemporary methods such as Glove (Pennington, Socher, and Manning 2014) and SkipGram Word2Vec (Mikolov et al. 2013) on the tasks of word-similarity and word-analogy. These tasks measure whether the representation of words learned from an unsupervised text corpus contains information about the semantic similarity between words or not.

A possible criticism of this choice of task for evaluation is that word-similarity and analogy do not represent an end-task. To that end, I will present experiments on the downstream tasks of Contextual Mention Retrieval and Entity Linking in Chapter 6. In this chapter I focus on the tasks of word similarity and analogy for two main reasons:

¹A previous version of this work was published in (Rastogi, Van Durme, and Arora 2015).

1. A large number of resources exist for extracting word co-occurrence matrices, and therefore it is possible to evaluate large-scale multi-view embeddings of words.
2. Words form the basis of language, and a large number of downstream NLP models benefit when initialized via word-embeddings, especially in low-resource scenarios. Therefore, the performance of a method on tasks such as analogy and similarity is important in its own right.

3.1 Introduction

(Winograd 1972) wrote that: “*Two sentences are paraphrases if they produce the same representation in the internal formalism for meaning*”. This intuition is made soft in vector-space models (Turney and Pantel 2010), where one says that expressions in language are paraphrases if their representations are *close* under some distance measure.

One of the earliest linguistic vector space models was Latent Semantic Analysis (LSA). LSA has been successfully used for Information Retrieval, but it is limited in its reliance on a single matrix, or *view*, of term co-occurrences. In this chapter, I address the single-view limitation of LSA by demonstrating that the framework of Generalized Canonical Correlation Analysis (GCCA) can be used to perform Multiview LSA (MVLSA). This approach allows for the use of an arbitrary number of views in the induction process, including embeddings induced using other algorithms. I also present a fast approximate method for performing GCCA and approximately recover the objective of (Pennington, Socher, and Manning 2014) while accounting for missing values.

My experiments show that MVLSA is competitive with state of the art approaches for inducing vector representations of words and phrases. As a methodological aside, I discuss the (in-)significance of conclusions being drawn from comparisons done on small sized datasets.

3.2 Motivation

LSA is an application of Principal Component Analysis (PCA) to a term-document cooccurrence matrix. The principal directions found by PCA form the basis of the vector space in which to represent the input terms (Landauer and Dumais 1997). A drawback of PCA is that it can leverage only a single source of data and it is sensitive to scaling.

An arguably better approach to representation learning is Canonical Correlation Analysis (CCA) that induces representations that are maximally *correlated* across two views, allowing the utilization of two distinct sources of data. While an improvement over PCA, being limited to only two views is unfortunate because many sources of data (perspectives) are frequently available in practice. In such cases, it is natural to extend CCA’s original objective of maximizing the correlation between two views by maximizing some measure of the matrix Φ that contains all the pairwise correlations between linear projections of the *covariates*. This is how Generalized Canonical Correlation Analysis (GCCA) was first derived by (Horst 1961). Recently these intuitive ideas about benefits of leveraging multiple sources of data have received strong theoretical backing due to work by (Sridharan and Kakade 2008) who showed that learning with multiple views is beneficial since it reduces the complexity of the learning problem by

restricting the search space. Recent work by (Anandkumar et al. 2014) showed that at least three views are necessary for recovering hidden variable models.

Note that there exist different variants of GCCA depending on the measure of Φ that one chooses to maximize. (Kettenring 1971) enumerated a variety of possible measures, such as the spectral-norm of Φ . Kettenring noted that maximizing this spectral-norm is equivalent to finding linear projections of the *covariates* that are most amenable to rank-one PCA, or that can be best explained by a single term factor model. This variant was named *MAX-VAR GCCA* and was shown to be equivalent to a proposal by (Carroll 1968), which searched for an auxiliary orthogonal representation G that was maximally correlated to the linear projections of the covariates. Carroll’s objective targets the intuition that representations leveraging multiple views should correlate with all provided views as much as possible.

3.3 Proposed Method: MVLSA

Let $X_j \in \mathbb{R}^{N \times d_j} \forall j \in [1, \dots, J]$ be the mean centered matrix containing data from view j such that row i of X_j contains the information for word w_i . Let the number of words in the vocabulary be N and number of contexts (columns in X_j) be d_j . Note that N remains the same and d_j varies across views. Following standard notation (Hastie, Tibshirani, and Friedman 2009) I call $X_j^\top X_j$ the scatter matrix and $X_j(X_j^\top X_j)^{-1}X_j^\top$ the projection matrix.

The objective of *MAX-VAR GCCA* can be written as the following optimization problem: Find $G \in \mathbb{R}^{N \times r}$ and $U_j \in \mathbb{R}^{d_j \times r}$ that solve:

$$\arg \min_{G, U_j} \sum_{j=1}^J \|G - X_j U_j\|_F^2 \quad (3.1)$$

$$\text{subject to } G^\top G = I.$$

The matrix G that solves problem (3.1) is my vector representation of the vocabulary. Finding G reduces to spectral decomposition of sum of projection matrices of different views: Define

$$P_j = X_j (X_j^\top X_j)^{-1} X_j^\top, \quad (3.2)$$

$$M = \sum_{j=1}^J P_j. \quad (3.3)$$

Then, for some positive diagonal matrix Λ , G and U_j satisfy:

$$MG = G\Lambda, \quad (3.4)$$

$$U_j = (X_j^\top X_j)^{-1} X_j^\top G. \quad (3.5)$$

The above expressions tell us that my word representations are the eigenvectors of the sum of J projection matrices. Also, note that the dimensions of G are orthogonal to each other. Orthogonality of representations can be a desirable property that I will discuss in more detail at the end of this chapter.

Computationally storing $P_j \in \mathbb{R}^{N \times N}$ is problematic owing to memory constraints. Further, the scatter matrices may be non-singular leading to an ill-posed procedure. I now describe a novel scalable GCCA with ℓ_2 -regularization to address these issues.

Approximate Regularized GCCA: GCCA can be regularized by adding $r_j I$ to scatter matrix $X_j^\top X_j$ before doing the inversion where r_j is a small constant

e.g. 10^{-8} . Projection matrices in (3.2) and (3.3) can then be written as

$$\tilde{P}_j = X_j(X_j^\top X_j + r_j I)^{-1} X_j^\top, \quad (3.6)$$

$$M = \sum_{j=1}^J \tilde{P}_j. \quad (3.7)$$

Next, to scale up GCCA to large datasets, I first form a rank- m approximation of projection matrices (Arora and Livescu 2012) and then extend it to an eigendecomposition for M following ideas by (Savostyanov 2014). Consider the rank- m SVD of X_j :

$$X_j = A_j S_j B_j^\top,$$

where $S_j \in \mathbb{R}^{m \times m}$ is the diagonal matrix with m -largest singular values of X_j and $A_j \in \mathbb{R}^{N \times m}$ and $B_j \in \mathbb{R}^{m \times d_j}$ are the corresponding left and right singular vectors. Given this SVD, write the j^{th} projection matrix as

$$\begin{aligned} \tilde{P}_j &= A_j S_j^\top (r_j I + S_j S_j^\top)^{-1} S_j A_j^\top, \\ &= A_j T_j T_j^\top A_j^\top, \end{aligned}$$

where $T_j \in \mathbb{R}^{m \times m}$ is a diagonal matrix such that $T_j T_j^\top = S_j^\top (r_j I + S_j S_j^\top)^{-1} S_j$.

Finally, I note that the sum of projection matrices can be expressed as $M = \tilde{M} \tilde{M}^\top$ where

$$\tilde{M} = [A_1 T_1 \dots A_J T_J] \in \mathbb{R}^{N \times mJ}.$$

Therefore, eigenvectors of matrix M , i.e. the matrix G that I am interested in finding, are the left singular vectors of \tilde{M} , i.e. $\tilde{M} = G S V^\top$. These left singular vectors can be computed by using Incremental PCA (Brand 2002) since \tilde{M} may be too large to fit in memory.

Let SVD_m denote a partial SVD where S_j is a rectangular diagonal matrix that contains only the m largest singular values and A_j, B_j are square, orthonormal, unitary matrices. Defining SVD_m like this ensures correctness but in practice one only needs to compute m columns of A_j . Take the SVD of X_j :

$$A_j S_j B_j^\top \xleftarrow{SVD_m} X_j$$

and substitute the above in equation 3.6 to get

$$\tilde{P}_j = A_j S_j^\top (r_j I + S_j S_j^\top)^{-1} S_j A_j^\top$$

. Define $T_j \in \mathbb{R}^{m \times m}$ to be the diagonal matrix such that $T_j T_j^\top = S_j^\top (r_j I + S_j S_j^\top)^{-1} S_j$ then

$$\tilde{P}_j = A_j T_j T_j^\top A_j^\top$$

. Now $\tilde{M} = [A_1 T_1 \dots A_J T_J] \in \mathbb{R}^{N \times mJ}$, then

$$M = \tilde{M} \tilde{M}^\top.$$

Performing QR decomposition of \tilde{M} gives

$$M = Q R R^\top Q$$

. Eigen decomposition of $R R^\top \in \mathbb{R}^{mJ \times mJ}$ results in eigen vectors U and eigen values S .

$$M = Q U S U^\top Q^\top$$

which implies $G = Q U$.

3.3.1 Computing SVD of mean centered X_j

Recall that I assumed X_j to be mean centered matrices. Let $Z_j \in \mathbb{R}^{N \times d_j}$ be sparse matrices containing mean-uncentered cooccurrence counts. Let $f_j = n_j \circ t_j$

be the preprocessing function that I will apply to Z_j :

$$Y_j = f_j(Z_j), \tag{3.8}$$

$$X_j = Y_j - 1(1^\top Y_j). \tag{3.9}$$

To compute the SVD of mean-centered matrices X_j I first compute the partial SVD of an uncentered matrix Y_j and then update it ((Brand 2006) provides details). I experimented with representations created from the uncentered matrices Y_j and found that they performed as well as the mean centered versions, but I will not mention them further since it is computationally efficient to follow the principled approach. I should note, however, that even the method of mean-centering the SVD produces an approximation.

3.3.2 Handling missing rows across views

With real data, it may happen that a term was not observed in a view at all. A large number of missing rows can corrupt the learned representations since the rows in the left singular matrix become zero. The procedure described above can not recover from this, and the representation for those words may become a one-hot vector. To counter this problem, I adopt a variant of the “missing-data passive” algorithm from (Van De Velden and Bijmolt 2006) who modified the GCCA objective to counter the problem of missing rows.² The objective now

²A more recent effort, by (Velden and Takane 2012), describes newer iterative and non-iterative (Test-Equating Method) approaches for handling missing values. It is possible that using one of those methods could improve performance.

becomes:

$$\arg \min_{G, U_j} \sum_{j=1}^J \left\| K_j (G - X_j U_j) \right\|_F^2 \quad (3.10)$$

$$\text{subject to } G^\top G = I,$$

where $[K_j]_{ii} = 1$ if row i of view j is observed and zero otherwise. Essentially K_j is a diagonal row-selection matrix which ensures that I will optimize the GCCA representations only on the observed rows. Note that $X_j = K_j X_j$ since the rows that K_j removed were already zero. Let, $K = \sum_j K_j$ then the optima of the objective can be computed by modifying equation (3.7) as:

$$M = K^{-\frac{1}{2}} \left(\sum_{j=1}^J P_j \right) K^{-\frac{1}{2}}. \quad (3.11)$$

Again, if I regularize and approximate the GCCA solution then I get G as the left singular vectors of $K^{-\frac{1}{2}} \tilde{M}$. I mean center the matrices using only the observed rows.

Also note that other heuristic weighting schemes could be used here. For example if I modify my objective as follows then I will approximately recover the objective of (Pennington, Socher, and Manning 2014):

$$\text{minimize: } \sum_{j=1}^J \left\| W_j K_j (G - X_j U_j) \right\|_F^2 \quad (3.12)$$

$$\text{subject to: } G^\top G = I$$

where

$$[W_j]_{ii} = \left(\frac{w_i}{w_{\max}} \right)^{\frac{3}{4}} \text{ if } w_i < w_{\max} \text{ else } 1,$$

$$\text{and } w_i = \sum_k [X_j]_{ik}.$$

3.4 Data

Training Data I used the English portion of the *Polyglot* Wikipedia dataset released by (Al-Rfou, Perozzi, and Skiena 2013) to create 15 *irredundant* views of co-occurrence statistics where element $[z]_{ij}$ of view Z_k represents that number of times word w_j occurred k words behind w_i . I selected the top 500K words by occurrence to create my vocabulary for the rest of the chapter. I lowercased all the words and discarded all words which were longer than 5 characters and contained more than 3 non-alphabetical symbols. This was done to preserve years and smaller numbers.

I extracted co-occurrence statistics from a large bitext corpus that was made by combining a number of parallel bilingual corpora as part of the ParaPhrase DataBase (PPDB) project: Table 3.1 gives a summary, (Ganitkevitch, Van Durme, and Callison-Burch 2013) provides further details. Element $[z]_{ij}$ of the *bitext* matrix represents the number of times English word w_i was automatically aligned to the foreign word w_j .

I also used the dependency relations in the *Annotated Gigaword Corpus* (Napoles, Gormley, and Van Durme 2012) to create 21 views³ where element $[z]_{ij}$ of view Z_d represents the number of times word w_j occurred as the governor of word w_i under dependency relation d .

I selected these dependency relations since they seemed to be particularly interesting which could capture different aspects of similarity.

I combined the knowledge of paraphrases present in FrameNet and PPDB

³Dependency relations employed: nsubj, amod, advmod, rcmmod, dobj, prep_of, prep_in, prep_to, prep_on, prep_for, prep_with, prep_from, prep_at, prep_by, prep_as, prep_between, xsubj, agent, conj_and, conj_but, pobj.

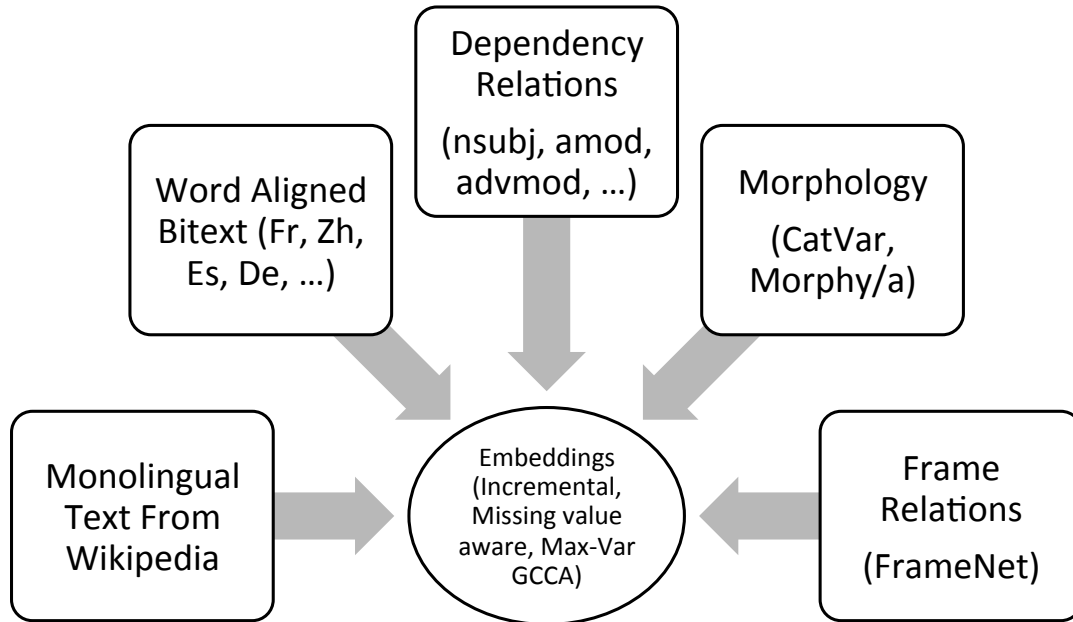


Figure 3.1: An illustration of datasets used.

by using the dataset created by (Rastogi and Van Durme 2014) to construct a *FrameNet* view. Element $[z]_{ij}$ of the *FrameNet* view represents whether word w_i was present in frame f_j . Similarly I combined the knowledge of morphology present in the *CatVar* database released by (Habash and Dorr 2003) and *morpha* released by (Minnen, Carroll, and Pearce 2001) along with *morphy* that is a part of WordNet. The morphological views and the frame semantic views were especially sparse with densities of 0.0003% and 0.03%. While the approach allows for an arbitrary number of distinct sources of semantic information, such as going further to include cooccurrence in WordNet synsets, I considered the described views to be representative, with further improvements possible as future work.

Test Data I evaluated the representations on the word similarity datasets listed in Table 3.2. The first 10 datasets in Table 3.2 were annotated with different

Language	Sentences	English Tokens
Bitext-Arabic	8.8M	190M
Bitext-Czech	7.3M	17M
Bitext-German	1.8M	44M
Bitext-Spanish	11.1M	241M
Bitext-French	30.9M	671M
Bitext-Chinese	10.3M	215M
Monotext-En-Wiki	75M	1700M

Table 3.1: Portion of data used to create GCCA representations (in millions).

rubrics and rated on different scales. However, broadly they all contain human judgments about how similar two words are. The “AN-SYN” and “AN-SEM” datasets contain 4-tuples of analogous words, and the task is to predict the missing word given the first three. Both of these are open vocabulary tasks while TOEFL is a closed vocabulary task.

Acronym	Size	$\sigma_{0.01}^{0.5}$	$\sigma_{0.01}^{0.7}$	$\sigma_{0.01}^{0.9}$	$\sigma_{0.05}^{0.5}$	$\sigma_{0.05}^{0.7}$	$\sigma_{0.05}^{0.9}$	Reference
MEN	3000	4.2	3.2	1.8	3.0	2.3	1.3	(Bruni et al. 2012)
RW	2034	5.1	3.9	2.3	3.6	2.8	1.6	(Luong, Socher, and Manning 2013)
SCWS	2003	5.1	4.0	2.3	3.6	2.8	1.6	(Huang et al. 2012)
SIMLEX	999	7.3	5.7	3.2	5.2	4.0	2.3	(Hill, Reichart, and Korhonen 2014)
WS	353	12.3	9.5	5.5	8.7	6.7	3.9	(Finkelstein et al. 2001)
MTURK	287	13.7	10.6	6.1	9.7	7.5	4.3	(Radinsky et al. 2011)
WS-REL	252	14.6	11.3	6.5	10.3	8.0	4.6	(Agirre et al. 2009)
WS-SEM	203	16.2	12.6	7.3	11.5	8.9	5.1	-Same-As-Above-
RG	65	28.6	22.3	12.9	20.6	16.0	9.2	(Rubenstein and Goodenough 1965)
MC	30	41.7	32.7	19.0	30.6	23.9	13.8	(Miller and Charles 1991)
AN-SYN	10675	-	-	0.95	-	-	0.68	(Mikolov et al. 2013)
AN-SEM	8869	-	-	1.03	-	-	0.74	-Same-As-Above-
TOEFL	80	-	-	8.13	-	-	6.63	(Landauer and Dumais 1997)

Table 3.2: List of test datasets used. The columns headed $\sigma_{p_0}^r$ contain *MRDS* values. The rows for accuracy based test sets contain σ_{p_0} which does not depend on r . See § 3.4.1 for details.

3.4.1 Significance of comparison

While surveying the literature, I found that performance on word similarity datasets is typically reported in terms of the Spearman correlation between the gold ratings and the cosine distance between normalized embeddings. However, researchers do not report measures of significance of the difference between the Spearman Correlations even for comparisons on small evaluation sets.⁴ This motivated me to define a method for calculating the *Minimum Required Difference for Significance (MRDS)*.

Minimum Required Difference for Significance (MRDS): Imagine two lists of ratings over the same items, produced respectively by algorithms A and B , and then a list of gold ratings T . Let r_{AT} , r_{BT} and r_{AB} denote the Spearman correlations between $A : T$, $B : T$ and $A : B$ respectively. Let \hat{r}_{AT} , \hat{r}_{BT} , \hat{r}_{AB} be their empirical estimates and assume that $\hat{r}_{BT} > \hat{r}_{AT}$ without loss of generality.

For word similarity datasets I define $\sigma_{p_0}^r$ as the MRDS, such that it satisfies the following proposition:

$$(r_{AB} < r) \wedge (|\hat{r}_{BT} - \hat{r}_{AT}| < \sigma_{p_0}^r) \implies pval > p_0$$

. Here $pval$ is the probability of the test statistic under the null hypothesis that $r_{AT} = r_{BT}$ found using the Steiger's test (Steiger 1980). The above constraint ensures that as long as the correlation between the competing methods is less than r and the difference between the correlations of the scores of the competing methods to the gold ratings is less than $\sigma_{p_0}^r$, then the p-value of the null hypothesis will be greater than p_0 . Now let us ask what is a reasonable upper bound on

⁴For example, the relative difference between competing algorithms reported by (Faruqui et al. 2014) could not be significant for the Word Similarity test set released by (Finkelstein et al. 2001), even if I assumed a correlation between competing methods as high as 0.9, with a p-value threshold of 0.05. Similar such comparisons on small datasets are performed by (Hill et al. 2014).

the agreement of ratings produced by competing algorithms: for instance, two algorithms correlating above 0.9 might not be considered meaningfully different. That leaves us with the second part of the predicate which ensures that as long as the difference between the correlations of the competing algorithms to the gold scores is less than $\sigma_{p_0}^r$ then the null hypothesis is more likely than p_0 .

I can find $\sigma_{p_0}^r$ as follows: Let *stest* denote Steiger’s test predicate which satisfies the following:

$$stest-p(\hat{r}_{AT}, \hat{r}_{BT}, r_{AB}, p_0, n) \implies pval < p_0$$

Once I define this predicate then I can use it to set up an optimistic problem where my aim is to find $\sigma_{p_0}^r$ by solving the following:

$$\sigma_{p_0}^r = \min\{\sigma | \forall 0 < r' < 1 \text{ } stest-p(r', \min(r' + \sigma, 1), r, p_0, n)\}$$

Note that MRDS is a liberal threshold and it only guarantees that differences in correlations below that threshold can never be statistically significant (under the given parameter settings). MRDS might optimistically consider some differences as significant when they are not, but it is at least useful in reducing some of the noise in the evaluations. The values of $\sigma_{p_0}^r$ are shown in Table 3.2.

For the accuracy based test-sets I found MRDS= σ_{p_0} that satisfied the following:

$$0 < (\hat{\theta}_B - \hat{\theta}_A) < \sigma_{p_0} \implies p(\theta_B \leq \theta_A) > p_0$$

Specifically, I calculated the posterior probability $p(\theta_B \leq \theta_A)$ with a flat prior of $\beta(1, 1)$ to solve the following:⁵ $\sigma_{p_0} = \min\{\sigma | \forall 0 < \theta < \min(1 - \sigma, 0.9) \text{ } p(\theta_B \leq \theta_A | \hat{\theta}_A = \theta, \hat{\theta}_B = \theta + \sigma, n) < p_0\}$ Here θ_A and θ_B are probability of correctness

⁵This instead of using McNemar’s test (McNemar 1947) since the Bayesian approach is tractable and more direct. A calculation with $\beta(0.5, 0.5)$ as the prior changed $\sigma_{0.5}$ from 6.63 to 6.38 for the TOEFL dataset but did not affect MRDS for the AN-SEM and AN-SYN datasets.

of algorithms A , B and $\hat{\theta}_A$, $\hat{\theta}_B$ are observed empirical accuracies.

Unfortunately, there are no widely reported train-test splits of the above datasets, leading to potential concerns of *soft supervision* (hyper-parameter tuning) on these evaluations throughout the existing literature. I report on the resulting impact of various parameterizations, and my final results are based on a single set of parameters used across all evaluation sets.

3.5 Experiments and Results

I wanted to answer the following questions through my experiments: (1) How do hyper-parameters affect performance? (2) What is the contribution of the multiple sources of data to performance? (3) How does the performance of MVLSA compare with other methods? I show the tuning runs on both larger and smaller datasets. I also highlight the top performing configurations in bold using the small threshold values in column $\sigma_{0.05}^{0.09}$ of Table 3.2.

Effect of Hyper-parameters f_j : I modeled the preprocessing function f_j as the composition of two functions, $f_j = n_j \circ t_j$. n_j represents nonlinear preprocessing that is usually employed with LSA. I experimented by setting n_j to be: identity; logarithm of count plus one; and the fourth root of the count.⁶ t_j represents the truncation of columns and can be interpreted as a type of regularization of the raw counts themselves through which I prune away the noisy contexts. The decrease in t_j also reduces the influence of views that have a large number of context columns and emphasizes the sparser views. Table 3.3 and Table 3.4 show the results.

⁶I also experimented with other powers of the counts (0.12, 0.5 and 0.75) on a smaller dataset and found that the fourth root performed the best.

Test Set	Log	Count	Count ^{$\frac{1}{4}$}
MEN	67.5	59.7	70.7
RW	31.1	25.3	37.8
SCWS	64.2	58.2	66.6
SIMLEX	36.7	27.0	38.0
WS	68.0	60.4	70.5
MTURK	57.3	55.2	60.8
WS-REL	60.4	52.7	62.9
WS-SEM	75.0	67.2	76.2
RG	69.1	55.3	75.9
MC	70.5	67.6	80.9
AN-SYN	45.7	21.1	53.6
AN-SEM	25.4	15.9	38.7
TOEFL	81.2	70.0	81.2

Table 3.3: Performance versus n_j , the non linear processing of cooccurrence counts. $t = 200K$, $m = 500$, $v = 16$, $k = 300$. All the top configurations determined by $\sigma_{0.05}^{0.09}$ are in bold font.

Test Set	6.25K	12.5K	25K	50K	100K	200K
MEN	70.2	71.2	71.5	71.6	71.2	70.7
RW	41.8	41.7	41.5	40.9	39.6	37.8
SCWS	67.1	67.3	67.1	67.0	66.9	66.6
SIMLEX	42.7	42.4	41.9	41.3	39.5	38.0
WS	68.1	70.8	71.6	71.2	70.2	70.5
MTURK	62.5	59.7	59.2	58.6	60.3	60.8
WS-REL	60.8	65.1	65.7	64.8	63.7	62.9
WS-SEM	77.8	78.8	78.8	78.2	76.5	76.2
RG	72.7	74.4	74.7	75.0	74.3	75.9
MC	75.2	75.9	79.9	80.3	76.9	80.9
AN-SYN	59.2	60.0	59.5	58.4	56.1	53.6
AN-SEM	37.7	38.6	39.4	39.2	38.4	38.7
TOEFL	88.8	87.5	85.0	83.8	83.8	81.2

Table 3.4: Performance versus the truncation threshold, t , of raw cooccurrence counts. I used $n_j = \text{Count}^{\frac{1}{4}}$ and other settings were the same as Table 3.3.

m : The number of left singular vectors extracted after SVD of the preprocessed cooccurrence matrices can again be interpreted as a type of regularization, since the result of this truncation is that I find cooccurrence patterns only between the top left singular vectors. I set $m_j = \max(d_j, m)$ with $m = [100, 300, 500]$. See table 3.5.

Test Set	100	200	300	500
MEN	65.6	68.5	70.1	71.1
RW	34.6	36.0	37.2	37.1
SCWS	64.2	65.4	66.4	66.5
SIMLEX	38.4	40.6	41.1	40.3
WS	60.4	67.1	69.4	71.1
MTURK	51.3	58.3	58.4	58.9
WS-REL	49.0	58.2	61.6	65.1
WS-SEM	73.6	76.8	76.8	78.0
RG	61.6	69.7	73.2	74.6
MC	65.6	74.1	78.3	77.7
AN-SYN	50.5	56.2	56.4	56.4
AN-SEM	24.3	31.4	34.3	40.6
TOEFL	80.0	81.2	82.5	80.0

Table 3.5: Performance versus m , the number of left singular vectors extracted from raw cooccurrence counts. I set $n_j = \text{Count}^{\frac{1}{4}}$, $t = 100K$, $v = 25$, $k = 300$.

k : Table 3.6 demonstrates the variation in performance versus the dimensionality of the learned vector representations of the words. Since the dimensions of the MVLSA representations are orthogonal to each other therefore creating lower dimensional representations is a trivial matrix slicing operation and does not require retraining.

v : Expression 3.12 describes a method to set W_j . I experimented with a different, more global, heuristic to set $[W_j]_{ii} = (K_{ww} \geq v)$, essentially removing all words that did not appear in v views before doing GCCA. Table 3.7 shows that changes in v are largely inconsequential for performance. In absence of clear

Test Set	10	50	100	200	300	500
MEN	49.0	67.0	69.7	70.2	70.1	69.8
RW	28.8	33.3	35.0	35.2	37.2	38.3
SCWS	57.8	64.4	65.2	66.1	66.4	65.1
SIMLEX	24.0	33.9	36.1	38.9	41.1	42.0
WS	46.8	63.4	69.5	69.5	69.4	66.0
MTURK	54.6	67.7	61.6	60.5	58.4	57.4
WS-REL	38.4	55.8	63.1	62.4	61.6	56.3
WS-SEM	55.3	69.9	76.9	77.1	76.8	75.6
RG	48.8	66.1	69.7	75.1	73.2	72.5
MC	37.0	59.0	71.3	79.1	78.3	75.7
AN-SYN	9.0	41.2	52.2	55.4	56.4	54.4
AN-SEM	2.5	21.8	34.8	35.8	34.3	33.8
TOEFL	57.5	72.5	76.2	81.2	82.5	85.0

Table 3.6: Performance versus k , the final dimensionality of the embeddings. I set $m = 300$ and other settings were same as Table 3.5.

evidence in favor of regularization I decided to regularize as little as possible and chose $v = 16$.

r_j : The regularization parameter ensures that all the inverses exist at all points in my method. I found that the performance of my procedure was invariant to r over a broad range from 1 to 1e-10. This was because even the 1000th singular value of my data was much higher than 1.

Contribution of different sources of data Table 3.8 shows an ablative analysis of performance where I remove individual views or some combination of them and measure the performance. It is clear by comparing the last column to the second column that adding in more views improves performance. Also I can see that the Dependency based views and the Bitext based views give a larger boost than the morphology and FrameNet based views, probably because the latter are so sparse. **Comparison to other word representation creation methods** There are a large number of methods of creating representations both

Test Set	16	17	21	25	29
MEN	70.4	70.4	70.2	70.1	70.0
RW	39.9	38.8	39.7	37.2	33.5
SCWS	67.0	66.8	66.5	66.4	65.7
SIMLEX	40.7	41.0	41.2	41.1	41.0
WS	69.5	69.4	69.5	69.4	69.1
MTURK	59.4	59.2	59.2	58.4	58.0
WS-REL	62.1	61.9	62.3	61.6	61.1
WS-SEM	76.8	76.8	77.0	76.8	76.8
RG	73.0	72.8	72.8	73.2	73.7
MC	75.0	76.0	76.5	78.3	78.6
AN-SYN	56.0	55.8	55.9	56.4	56.0
AN-SEM	34.6	34.3	34.0	34.3	34.3
TOEFL	85.0	85.0	83.8	82.5	80.0

Table 3.7: Performance versus minimum view support threshold v , The other hyper-parameters were $n_j = \text{Count}^{\frac{1}{4}}$, $m = 300$, $t = 100K$. Though a clear best setting did not emerge, I chose $v = 25$ as the middle ground.

multilingual and monolingual. There are many new methods such as by (Yu and Dredze 2014), (Faruqui et al. 2014), (Hill and Korhonen 2014), and (Weston, Chopra, and Adams 2014) that are performing multiview learning and could be considered here as baselines: however it is not straight-forward to use those systems to handle the variety of data that I am using. Therefore, I directly compare my method to the Glove and the SkipGram model of Word2Vec as the performance of those systems is considered state of the art. I trained these two systems on the English portion of the *Polyglot* Wikipedia dataset.⁷ I also combined their outputs using MVLSA to create *MV-G-WSG* embeddings.

I trained my best MVLSA system with data from all views and by using the individual best settings of the hyper-parameters. Specifically the configuration I

⁷I explicitly provided the vocabulary file to Glove and Word2Vec and set the truncation threshold for Word2Vec to 10. Glove was trained for 25 iterations. Glove was provided a window of 15 previous words, and Word2Vec used a symmetric window of 10 words.

Test Set	All Views	!Framenet	!Morphology	!Bitext	!Wikipedia	!Dependency	!Morphology !Framenet	!Morphology !Framenet !Bitext
MEN	70.1	69.8	70.1	69.9	46.4	68.4	69.5	68.4
RW	37.2	36.4	36.1	32.2	11.6	34.9	34.1	27.1
SCWS	66.4	65.8	66.3	64.2	54.5	65.5	65.2	60.8
SIMLEX	41.1	40.1	41.1	37.8	32.4	44.1	38.9	34.4
WS	69.4	69.1	69.2	67.6	43.1	70.5	69.3	66.6
MTURK	58.4	58.3	58.6	55.9	52.7	59.8	57.9	55.3
WS-REL	61.6	61.5	61.4	59.4	38.2	63.5	62.5	58.8
WS-SEM	76.8	76.3	76.7	75.9	48.1	75.7	75.8	73.1
RG	73.2	72.0	73.2	73.7	45.0	70.8	71.9	74.0
MC	78.3	75.7	78.2	78.2	46.5	77.5	76.0	80.2
AN-SYN	56.4	56.3	56.2	51.2	37.6	50.5	54.4	46.0
AN-SEM	34.3	34.3	34.3	36.2	4.1	35.3	34.5	30.6
TOEFL	82.5	82.5	82.5	71.2	45.0	85.0	82.5	65.0

Table 3.8: Performance versus views removed from the multiview GCCA procedure. !Framenet means that the view containing counts derived from Frame semantic dataset was removed. Other columns are named similarly. The other hyperparameters were $n_j = \text{Count}^{\frac{1}{4}}$, $m = 300$, $t = 100K$, $v = 25$, $k = 300$.

used was as follows: $n_j = \text{Count}^{\frac{1}{4}}$, $t = 12.5K$, $m = 500$, $k = 300$, $v = 16$. To make a fair comparison, I also provide results where I used only the views derived from the *Polyglot* Wikipedia corpus. See column *MVLSA (All Views)* and *MVLSA (Wiki)* respectively. It is visible that MVLSA on the monolingual data itself is competitive with Glove but worse than Word2Vec on the word similarity datasets and it is substantially worse than both the systems on the AN-SYN and AN-SEM datasets. However with the addition of multiple views, MVLSA makes substantial gains, shown in column *MV Gain*, and after consuming the Glove and WSG embeddings, it again improves performance by some margins, as shown in column *G-WSG Gain*, and outperforms the original systems. Using GCCA itself for system combination provides closure for the MVLSA algorithm

since multiple distinct approaches can now be simply fused using this method. Finally, I contrast the Spearman correlations r_s with Glove and Word2Vec before and after including them in the GCCA procedure. The values demonstrate that including Glove and WSG during GCCA increased the correlation between them and the learned embeddings, which supports my motivation for performing GCCA in the first place.

Test Set	Glove	WSG	MV G-WSG	MVLSA Wiki	MVLSA All Views	MVLSA Combined	MV Gain	G-WSG Gain	r_s MVLSA Glove WSG	r_s MV-G-WSG Glove WSG
MEN	70.4	73.9	76.0	71.4	71.2	75.8	-0.2	4.6 [†]	71.9 89.1	85.8 99.1
RW	28.1	32.9	37.2	29.0	41.7	40.5	12.7 [†]	-1.2	72.3 74.2	80.2 79.3
SCWS	54.1	65.6	60.7	61.8	67.3	66.4	5.5 [†]	-0.9	87.1 94.5	91.3 90.6
SIMLEX	33.7	36.7	41.1	34.5	42.4	43.9	7.9 [†]	1.5	62.4 78.2	79.3 80.1
WS	58.6	70.8	67.4	68.0	70.8	70.1	2.8 [†]	-0.7	72.3 88.1	81.8 91.4
MTURK	61.7	65.1	59.8	59.1	59.7	62.9	0.6	3.2	80.0 87.7	87.3 99.1
WS-REL	53.4	63.6	59.6	60.1	65.1	63.5	5.0 [†]	-1.6	58.2 81.0	69.6 80.1
WS-SEM	69.0	78.4	76.1	76.8	78.8	79.2	2.0	0.4	74.4 90.6	83.9 94.1
RG	73.8	78.2	80.4	71.2	74.4	80.8	3.2	6.4 [†]	80.3 90.6	91.8 99.1
MC	70.5	78.5	82.7	76.6	75.9	77.7	-0.7	2.8	80.1 94.1	91.4 99.1
AN-SYN	61.8	59.8	51.0	42.7	60.0	64.3	17.3 [†]	4.3 [†]		
AN-SEM	80.9	73.7	73.5	36.2	38.6	77.2	2.4 [†]	38.6 [†]		
TOEFL	83.8	81.2	86.2	78.8	87.5	88.8	8.7 [†]	1.3		

Table 3.9: Comparison of Multiview LSA against Glove and WSG(Word2Vec Skip Gram). Using $\sigma_{0.05}^{0.9}$ as the threshold I highlighted the top performing systems in bold font. [†] marks significant increments in performance due to use of multiple views in the *Gain* columns. The r_s columns demonstrate that GCCA increased Pearson correlation.

3.6 Previous Work

Vector space representations of words have been created using diverse frameworks including Spectral methods (Dhillon, Foster, and Ungar 2011; Dhillon et al. 2012),

⁸ Neural Networks (Mikolov, Yih, and Zweig 2013; Collobert and Lebrete 2013),

⁸cis.upenn.edu/~ungar/eigenwords

and Random Projections (Ravichandran, Pantel, and Hovy 2005; Bhagat and Ravichandran 2008; Chan, Callison-Burch, and Van Durme 2011).⁹ They have been trained using either one (Pennington, Socher, and Manning 2014)¹⁰ or two sources of cooccurrence statistics (Zou et al. 2013; Faruqui and Dyer 2014; Bansal, Gimpel, and Livescu 2014; Levy and Goldberg 2014)¹¹ or using multi-modal data (Hill and Korhonen 2014; Bruni et al. 2012).

(Dhillon, Foster, and Ungar 2011) and (Dhillon et al. 2012) were the first to use CCA as the primary method to learn vector representations and (Faruqui and Dyer 2014) further demonstrated that incorporating bilingual data through CCA improved performance. More recently this same phenomenon was reported by (Hill et al. 2014) through their experiments over neural representations learned from MT systems. Outside of the NLP community (Sun, Priebe, and Tang 2013; Tripathi 2011) are examples of works that have used GCCA for “data fusion”. Various other researchers have tried to improve the performance of their paraphrase systems or vector space models by using diverse sources of information such as bilingual corpora (Bannard and Callison-Burch 2005; Huang et al. 2012; Zou et al. 2013),¹² structured datasets (Yu and Dredze 2014; Faruqui et al. 2014) or even tagged images (Bruni et al. 2012). However, most previous work¹³ did not adopt the general, simplifying view that all of these sources of data

⁹code.google.com/p/word2vec/metaoptimize.com/projects/wordreprs

¹⁰nlp.stanford.edu/projects/glove

¹¹ttic.uchicago.edu/~mbansal/data/syntacticEmbeddings.zip, cs.cmu.edu/~mfaruqui/soft.html

¹²An example of complementary views: (Chan, Callison-Burch, and Van Durme 2011) observed that monolingual distributional statistics are susceptible to conflating antonyms, where bilingual data is not; on the other hand, bilingual statistics are susceptible to noisy alignments, where monolingual data is not.

¹³(Ganitkevitch, Van Durme, and Callison-Burch 2013) did employ a rich set of diverse cooccurrence statistics in constructing the initial PPDB, but without a notion of “training” a joint representation beyond random projection to a binary vector subspace (bit-signatures).

are just cooccurrence statistics coming from different sources with underlying latent factors.¹⁴

(Bach and Jordan 2005) presented a probabilistic interpretation for CCA. Though they did not generalize it to include GCCA, I believe that one could give a probabilistic interpretation of *MAX-VAR GCCA*. Such a probabilistic interpretation would allow for an online-generative model of lexical representations, which unlike methods like Glove or LSA would allow us to naturally perplexity or generate sequences. I also note that (Vía, Santamaría, and Pérez 2007) presented a neural network model of GCCA and adaptive/incremental GCCA. To the best of my knowledge, both of these approaches have not been used for word representation learning.

CCA is also an algorithm for multi-view learning (Kakade and Foster 2007; Ganchev et al. 2008) and when I view my work as an application of multiview learning to NLP, this follows a long chain of effort started by (Yarowsky 1995) and continued with *Co-Training* (Blum and Mitchell 1998), *CoBoosting* (Collins and Singer 1999) and *2 view perceptrons* (Brefeld et al. 2006).

3.7 Conclusion

This chapter is based on the following published paper:

Rastogi, Pushpendre, Benjamin Van Durme, and Raman Arora (2015).

“Multi- view LSA: Representation Learning Via Generalized CCA”. In:

Proceedings of NAACL.

¹⁴Note that while (Faruqui et al. 2014) performed belief propagation over a graph representation of their data, such an undirected weighted graph can be viewed as an adjacency matrix, which is then also a co-occurrence matrix.

The main ideas and scientific contribution of this chapter are:

- The first to construct word embeddings from massively multi-view datasets.
- The first algorithm for scaling Generalized CCA to large datasets via a novel approximation technique.
- A new procedure, *MRDS*, for measuring the significance of results, based only on the spearman-correlation values and dataset size.

While previous efforts demonstrated that incorporating two views is beneficial in word-representation learning, I extended that thread of work to a logical extreme and created *MVLSA* to learn distributed representations using data from 46 views!¹⁵ Through evaluation of my induced representations, shown in Table 3.9, I demonstrated that the MVLSA algorithm could leverage the information present in multiple data sources to improve performance on a battery of tests against state of the art baselines. To perform MVLSA on large vocabularies with up to 500K words, I presented a fast, scalable algorithm. I also showed that a close variant of the Glove objective proposed by (Pennington, Socher, and Manning 2014) could be derived as a heuristic for handling missing data under the MVLSA framework. To better understand the benefit of using multiple sources of data, I performed MVLSA using views derived only from the monolingual Wikipedia dataset thereby providing a more principled alternative of LSA that removes the need for heuristically combining word-word cooccurrence matrices into a single matrix. Finally, while surveying the literature I noticed that not enough emphasis was being given towards establishing the significance of

¹⁵Code and data available at www.cs.jhu.edu/~prastog3/mvlsa

comparative results and proposed a method, (*MRDS*), to filter out insignificant comparative gains between competing algorithms.

Future Work Column *MVLSA Wiki* of Table 3.9 shows us that MVLSA applied to monolingual data has mediocre performance compared to the baselines of Glove and Word2Vec on word similarity tasks and performs surprisingly worse on the AN-SEM dataset. I believe that the results could be improved by (1) either using recent methods for handling missing values mentioned in footnote 2 or by using the heuristic count dependent non-linear weighting mentioned by (Pennington, Socher, and Manning 2014) and that sits well within my framework as exemplified in Expression 3.12 (2) by using even more views, which look at the future words as well as views that contain PMI values. Finally, I note that Table 3.8 shows that certain datasets can actually degrade performance over certain metrics. Therefore I am exploring methods for performing discriminative optimization of weights assigned to views, for purposes of task-based customization of learned representations.

Chapter 4

Neural Variational Set Expansion

People use words and sentences to communicate with each other about real-world entities. In the previous chapter, I presented a “shallow” algorithm MVLSA for learning representations of words. In this chapter, I go deeper and present a novel “deep” representation learning method for learning representations of entities grounded in natural language text.¹ For this chapter an entity is a set of mentions across multiple documents that refer to the same real-world object. Distributed representations of such mention-sets can aid Information extraction and retrieval systems. To that end, I focus on the task of *Entity Recommendation*. Many existing information retrieval systems that operate on entities rely on clean, manually curated sets of entities for their operation. Because users often work with unclean, automatically generated KGs and require interpretable tools; therefore, they are often unable to incorporate such algorithms in their workflow fully. I propose Neural Variational Set Expansion to extract actionable information from a noisy knowledge graph (KG) grounded in natural language and also

¹A previous version of this work was published in (Rastogi et al. [2018](#)).

propose a general approach for increasing the interpretability of recommendation systems.

Akin to prior entity-focused retrieval definitions, a query consists of one or more entities, with the intent of retrieving similar entities. Differing from prior work, I focus neither on manually curated knowledge bases, nor collections of entity-labeled documents such as Wikipedia. I demonstrate the usefulness of applying a variational autoencoder to the Entity Set Expansion task based on a realistic automatically generated KG. Further, I describe an approach for ESE, Neural Variational Set Expansion, which supports humanly interpretable query rationales, and outperforms baselines such as Bayesian Sets and BM25.

4.1 Introduction

Imagine a physician trying to pinpoint a specific diagnosis or a security analyst attempting to uncover a terrorist network. In both scenarios, a *domain expert* may try to find answers based on prior known, relevant entities – either a list of diagnoses of with similar symptoms that a patient is experiencing or a list of known conspirators. Instead of manually looking for connections between potential answers and prior knowledge, a *searcher* would like to rely on an automatic *Recommender* to find the connections and answers for them, i.e., related entities.

In the information retrieval (IR) community, Entity Set Expansion (ESE) is the established task of recommending entities that are similar to a provided seed of entities.² ESE has been applied in Question Answering (Wang et al. 2008),

²I refer to the items in the seed as entities, but they can also be referred to as items or elements

Relation Extraction (Lang and Henderson 2013) and Information Extraction (He and Grishman 2015) settings. The physician and journalist in my example cannot fully take advantage of IR advances in ESE for two main reasons. Recent advances 1) often assume access to a clean, large Knowledge Graph and 2) are uninterpretable.

Many advanced ESE algorithms rely on manually curated, clean Knowledge Graphs (KG), e.g. DBpedia (Auer et al. 2007) and Freebase (Bollacker et al. 2008). In clean KGs duplicate entities are merged, entities rarely are isolated, and entities with similar names are properly disambiguated. However, in real-world settings, users do not always have access to clean KGs, and instead, they may rely on automatically generated KGs. Such KGs are often *noisy* because they are created from complicated and error-prone NLP processes – illustrated in Figure 4.1. For example, automatic KGs may include duplicate entities, associations (relations) between entities may be missing, and entities with similar names may be incorrectly disambiguated. Similarly, faulty coreference or entity linking may fail to merge duplicate entities, may create many isolated entities, and may poorly disambiguate entities with similar names. These imperfections prevent machine learning approaches from performing well on automatically generated KGs. Furthermore, many ESE algorithm’s performance degrades as the sparsity and unreliability of KGs increases (Pujara, Augustine, and Getoor 2017; Rastogi, Lyzinski, and Van Durme 2017). Therefore, in practice, users working with large KGs even now only rely on weighted boolean and keyword searches (Jin, French, and Michel 2005; Gadepally et al. 2016) instead of advanced KG completion algorithms

Advanced ESE methods, especially those that rely on neural networks, are

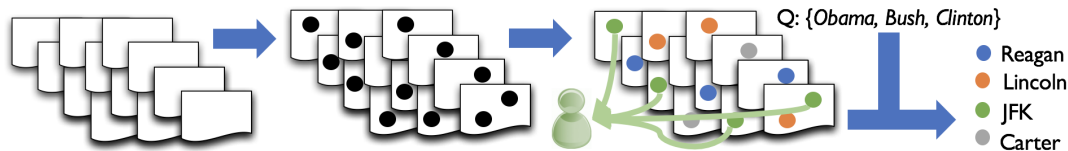


Figure 4.1: My *Entity Set Expansion* (ESE) system assumes a corpus that has been labeled with entity mentions which are clustered via cross-document co-reference and linking to a knowledge base; together known as *entity discovery and linking* (EDL). Given a query containing *Obama*, *Bush*, and *Clinton*, the ESE system returns other U.S. presidents found in the KG.

uninterpretable (Mitra and Craswell 2017). If a physician can not explain decisions, patients may not trust her, and if a journalist can not demonstrate how a certain individual is acting unethically or above the law, a resulting article may lack credibility. Furthermore, uninterpretable may limit the applications of advancements in IR, and more broadly artificial intelligence, as humans “won’t trust an A.I. unless it can explain itself.”³

I introduce Neural Variational Set Expansion (NVSE) to advance the applicability of ESE research. NVSE is an unsupervised model based on Variational Autoencoders (VAEs) that receives a query, consisting of a small set of entities and uses a Bayesian approach to determine a latent concept that unifies entities in the query, and returns a ranked list of similar entities based on the previously determined unified latent concept. I refer to my method as Neural Variational Set Expansion since NVSE uses a VAE to model the latent concept as a Gaussian distributed random variable for the task of Set Expansion.

NVSE does not require supervised examples of queries and responses, nor a manually curated KG. It also does not require nor pre-built clusters of entities. Instead, my method only requires sentences with linked entity mentions, i.e.,

³<https://nyti.ms/2hR1S15>

spans of tokens associated with a KG entity, often included in automatically generated KGs.

NVSE is robust to noisy automatically generated KGs, thus removing the need to rely on manually curated, clean KGs. I evaluate NVSE on the ESE task using Tinkerbelle (Al-Badrashiny et al. 2017), an automatically generated KG that placed first in the TAC KGP shared-task. Unlike how ESE has been used to improve entity linking for KG construction (Gottipati and Jiang 2011), my goal is the opposite: I leverage noisy automatically generated KGs to perform ESE. NVSE is interpretable; it outputs **query rationales** – a summarization of features the model associated with the query – and **result justifications** – an ordered list of sentences from the underlying corpus that justify why my method returned that entity. Query rationales and result justifications are reminiscent of *annotator rationales* (Zaidan, Eisner, and Piatko 2007).

To my knowledge, this is the first unsupervised neural approach for ESE as opposed to neural methods for supervised collaborative filtering (Lee, Song, and Moon 2017). All code and data is available at github.com/se4u/nvse and a video demonstration of the system is available at youtu.be/sg0_wvuP1zM.

4.2 Related Work

Methods dependent on external information. Since automatically generated KGs can be noisy, some methods utilize information beyond entity links and mentions to aid ESE. (Paşca and Van Durme 2007) use search engine query logs to extract attributes related to entities and (Paşca and Van Durme 2008) extract sets of instances associated with class labels based on web documents

and queries. (Pantel et al. 2009) use a large amount of web data as they apply a learned word similarity matrix extracted from a 200 billion word Internet crawl to the ESE task. Both (He and Xin 2011)’s SEISA system and (Tong and Dean 2008)’s Google Sets use lists of items from the Internet and try to determine which elements in the lists are most relevant to a query. (Sadamitsu et al. 2011a) rely on given topic information about the queried entities to train a discriminative system. More recent approaches also use external information. (Zaheer et al. 2017) use LDA (Blei, Ng, and Jordan 2003) to create word clusters for supervision, and (Vartak et al. 2017) use manual annotations by Twitter users. (Zheng et al. 2017) uses inter-entity links in knowledge graphs which are very sparse in automatically generated KGs (Pujara, Augustine, and Getoor 2017; Rastogi, Lyzinski, and Van Durme 2017). All of these approaches use more information than just entity links and mentions.

Methods for comparing entities. Set Expander for Any Language (SEAL) (Wang and Cohen 2007) and its variants (Wang and Cohen 2008; Wang and Cohen 2009) learn similarities between new words and example words using methods like Random Walks and Random Walks With Restart. Similar to (Lin 1998)’s using cosine and Jaccard similarity to find similar words, SEISA uses these metrics to expand sets. These methods are limited to only extracting words that co-occur. Because they are applied to web-scale data, SEAL and SEISA assume entities will eventually co-occur. This assumption might not be valid in an underlying corpus used to generate a KG automatically. In contrast to those approaches, NVSE finds similar entities based on a kernel between distributions.

Queries as natural language. In the INEX-XER shared-task, queries

were represented as natural language questions (Demartini, Iofciu, and De Vries 2010). (Metzger, Schenkel, and Sydow 2014) and (Zhang et al. 2017) propose methods to extract related entities in a KG based on a natural language query. This scenario is similar to a person interacting with a system like Amazon Alexa. However, my setup better reflects users searching for similar entities in a KG as it is more efficient for users to type entities of interest instead of natural language text.

Neural Collaborative Filtering. I am not the first to incorporate neural methods in a recommendation system. Recently, (He et al. 2017) and (Lee, Song, and Moon 2017) presented deep auto-encoders for collaborative filtering. Collaborative Filtering assumes a large dataset of previous user interactions with the search engine. For many domains, it is not possible to create such a dataset since new data is added every day and queries change rapidly based on different users and domains. Therefore, I propose the first neural method which does not use supervision for Entity Set Expansion. (Li and She 2017) use a citation dataset and their recommendations only include users with less than ten articles. They only gave recommendations for entities that appeared in at least 10 articles in the corpus.

Unsupervised Clustering for Entity Resolution (Sadamitsu et al. 2011b) proposed to learn the latent topics of documents for alleviating problems of “semantic drift” in Entity Set Expansion. Semantic drift refers to the common problem faced by entity set expansion algorithms of changes in the extraction criteria. In order to combat this problem they modeled the latent topics with LDA (Latent Dirichlet Allocation) (Blei, Ng, and Jordan 2003) and utilized the topic information in three ways:

- First they used the topic distribution of documents to generate features for their set expansion system.
- Second, they selected negative examples for training a discriminative system.
- And third they pruned certain examples in their iterative training method.

Instead of creating a pipelined approach using a pre-existing topic model, our approach allows us to create a topic model that can be trained end-to-end and which is directly amenable to learning non-linear features of the data. A similarity between the two methods is that variational inference can be used for learning the parameters of both the models.

4.3 Notation

Let \mathcal{D} be the corpus of documents and \mathcal{V} be the vocabulary of tokens that appear in \mathcal{D} . I define a document as a sequence of sentences and a sentence as a sequence of tokens. Let \mathcal{X} be the set of entities discovered in \mathcal{D} and I refer to its size as X . Each entity $x \in \mathcal{X}$ is linked to the tokens that mention x .⁴ Let \mathcal{V}' be the set of tokens linked to any $x \in \mathcal{X}$, and let \mathcal{M}_x be the multiset of sentences that mention x in the corpus. For example, consider an entity named “Batman” and a document containing three sentences {Batman is good., He is smart. Life is good.}. “Batman” is linked to tokens Batman and He,

In ESE, a system receives query \mathcal{Q} – a subset of \mathcal{X} – and has to sort the elements remaining in $\mathcal{R} = \mathcal{X} \setminus \mathcal{Q}$. The elements that are most similar to \mathcal{Q}

⁴I ignore confidence scores that entity linking systems often assign to a link because confidence scores will prevent us from using a multinomial distribution to model a document as a bag-of-words.

should appear higher in the sorted order and elements dissimilar to \mathcal{Q} should be ranked lower.

4.4 Baseline Methods

Before introducing NVSE, I describe the four baselines systems: BM25, Bayesian Sets, Word2Vecf, and SetExpan. I do not compare to DeepSets (Zaheer et al. 2017), as it is a supervised method that requires entity clusters.

For each x , I create a feature vector $f_x \in \mathbb{Z}^F$ from \mathcal{M}_x , by concatenating three vectors that count how many times 1) a token in \mathcal{V} appeared in \mathcal{M}_x 2) a document in \mathcal{D} mentioned x and 3) a token in \mathcal{V}' appeared in \mathcal{M}_x . Thus, $F = V + D + V'$.

4.4.1 BM25

Best Match 25 (BM25) is “one of the most successful text-retrieval algorithms” (Robertson and Zaragoza 2009).⁵ BM25 ranks remaining entities in \mathcal{R} according to the score function

$$\text{score}_{BM}(\mathcal{Q}, x) = \sum_{i=1}^F \frac{\text{IDF}[i] f_x[i] \bar{f}_{\mathcal{Q}}[i] (k_1 + 1)}{f_x[i] + k_1 (1 - b + b \sum_j f_x[j] / \bar{L})},$$

where $f_x[j]$ denotes the j -th feature value in f_x , $\bar{f}_{\mathcal{Q}}$ is the sum of $f_x \forall x \in \mathcal{Q}$ and \mathbb{I} is the indicator function. k_1 and b are hyperparameters that commonly set to 1.5 and 0.75 (Manning, Raghavan, and Schütze 2008). \bar{L} is the average total count of a feature in the entire corpus and $\text{IDF}[i]$ is the inverse document

⁵Lucene replaced tf-idf with BM25 as its default algorithm: <https://issues.apache.org/jira/browse/LUCENE-6789>

frequency of the i^{th} feature. They are computed as,

$$\begin{aligned}\bar{L} &= \sum_{x \in \mathcal{X}} \sum_j f_x[j] / X \\ \text{IDF}[i] &= \log \frac{X - \text{DF}[i] + 0.5}{\text{DF}[i] + 0.5} \\ \text{DF}[i] &= \sum_{x \in \mathcal{X}} \mathbb{I}[f_x[i] > 0].\end{aligned}$$

4.4.2 Bayesian Sets

(Ghahramani and Heller 2006) introduced the Bayesian Sets (BS) method which converts ESE into a bayesian model selection problem. BS compares the probabilities that the query entities are generated from a single sample of a latent variable $z \in \Delta^F$ with the probability that the entities were generated from independent samples. Δ^F is the $F - 1$ dimensional probability simplex. Note that z has the same dimensionality as the observed features. Given \mathcal{Q} and π , the prior distribution of z , BS infers the posterior distribution of z , $p(z|\mathcal{Q})$, and computes the following score

$$\text{score}_{BS}(\mathcal{Q}, x) = \log \frac{E_{p(z|\mathcal{Q})}[p(x|z)]}{E_{\pi(z)}[p(x|z)]}. \quad (4.1)$$

(Ghahramani and Heller 2006) computed score_{BS} in close form by selecting the conditional probability, $p(x|z)$, from an exponential family distribution and setting π to be its conjugate prior. They showed that if $p(x|z)$ is multivariate Bernoulli then BS requires a single matrix multiplication (Appendix A.1) and I use this setting for my experiments.

4.4.3 Word2Vecf

(Levy and Goldberg 2014) generalize (Mikolov et al. 2013)’s Skip-Gram model as Word2Vecf to include arbitrary contexts. I embed entities with Word2Vecf by using the entity IDs as words ⁶ and the tokens in the sentences mentioning those entities as contexts. Note that all tokens in the sentence, except for some stop words, are used as contexts and not just co-occurrent entities. I rank the entities in the order of their total distance to the entities in the query set as

$$\text{score}_{W2V}(\mathcal{Q}, x) = - \sum_{\tilde{x} \in \mathcal{Q}} (v_x - v_{\tilde{x}})^2. \quad (4.2)$$

Here, v_x represents the L2-normalized embedding for x .

4.4.4 SetExpan

(Shen et al. 2017) introduce SetExpan, a SOTA framework combining context feature selection with ranking ensembles, for set expansion. SetExpan outperformed other SE methods such as SEISA in their evaluation. SetExpan represents entities by the contexts that they are mentioned in. For example, the context features for Batman from § 4.3 will be {__ is good, __ is smart}. The contexts are used to create a large feature vector which can be used to compute the inter-entity similarity. The authors argue that using all possible features for computing entity similarity can lead to overfitting and semantic drift. To combat these problems, SetExpan builds the entity set iteratively by cycling between a context feature selection step and an entity selection step. In context feature selection, each context feature is assigned a score based on the set of currently expanded entities. Based on these scores, the context-features are reranked, and

⁶Converting entity mentions to entity IDs allows us to overcome issues related to embedding multi-word expressions as explained in (Poliak et al. 2017).

the top few context features are selected. The entity selection proceeds by the bootstrap sampling of the chosen context features and using those features to create multiple different ranked lists of entities. Multiple different ranked lists are finally combined via a heuristic method for ensembling different ranked lists to create a new set of expanded entities. This process is repeated to convergence to get the final list of expanded entities.

4.5 Neural Variational Set Expansion

Like BS, Neural Variational Set Expansion first determines the underlying concept, or topic, underlying the query and then ranks entities based on that concept. My method differs from BS because I use a deep generative model with a low dimensional concept representation, to simulate how a concept may generate a query. Also I use a “distance” (§ 4.5.2) between posterior distributions for ranking entities in lieu of bayesian model comparison.

4.5.1 Inference Step 1: Concept Discovery

My model (Fig. 4.2) is as follows: $z \in \mathbb{R}^d$ is a low dimensional latent gaussian random variable representing the concept of a query. z is sampled from a fixed prior distribution $\pi = \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, i.e. $z \sim \pi$. The members of \mathcal{Q} are sampled conditionally independently given z . z is mapped via a multi layer perceptron (MLP), called $\text{NN}_\theta^{(g)}$, to g , the p.m.f. of a multinomial distribution that generates f_x , the features of x . $\text{NN}_\theta^{(g)}$ is a neural network with a softmax output layer and parameters θ . $f_x \in \mathbb{Z}^F$ are sampled i.i.d. from $p(f|z, \theta) = \text{NN}_\theta^{(g)}(z)$.⁷

⁷My generative model is inspired by (Miao, Yu, and Blunsom 2016)’s NVDM. They assume that a single latent variable generates only one observation, but I posit that the same latent variable z generates all observations in \mathcal{Q} .

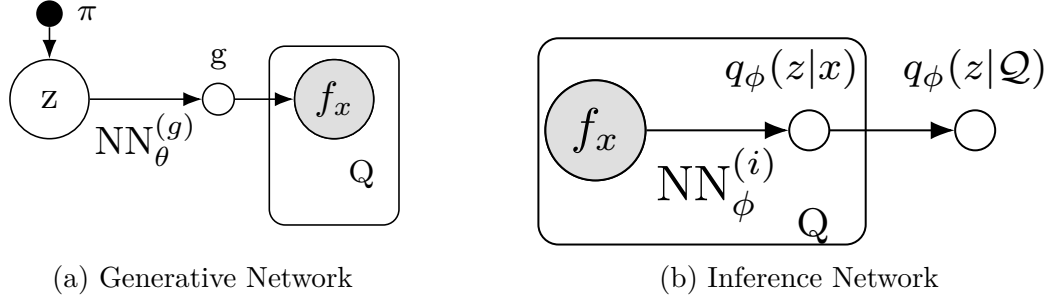


Figure 4.2: The generative model of query generation is on the left and the variational inference network is on the right. Small nodes denote probability distributions, gray nodes are observations and the black node π is the known prior. $\text{NN}_\theta^{(g)}$ transforms z to g and the $\text{NN}_\phi^{(i)}$ transforms f_x to $q_\phi(z|x)$.

In other words, the vector f_x contains the counts of observed features for x that were sampled from g , and g was itself sampled by passing a Gaussian random variable through a neural network.

Under this deep-generative model, a concept vector can simultaneously trigger multiple observed features. This allows us to capture the correlations amongst features triggered by a concept. For example, the concept of **president** can simultaneously trigger features such as white house, executive order, or airforce one.

To infer the latent variable z ideally, I should compute $p_\theta(z|Q)$, the posterior distribution of z given the observations Q . Unfortunately, this computation is intractable because the prior is not conjugate to the likelihood that has a neural network. Another problem is that it is unrealistic to assume access to a large set of ESE queries at training time, because user’s information needs keep changing; therefore the approach used by (Zaheer et al. 2017) in DeepSets to discriminatively learn a neural scoring function is *impractical* for set expansion. For the same reason, it is also not possible to learn a single neural network whose

input is \mathcal{Q} and which directly approximates $p_\theta(z|\mathcal{Q})$. Therefore it is non-trivial to apply the VAE framework to ESE. To overcome these problems I make the assumption that a query \mathcal{Q} is conjunctive in nature, i.e. if entity x_1 and x_2 are present in \mathcal{Q} then results that are relevant to *both* x_1 and x_2 simultaneously should be given a higher ranking than results that are related to x_1 but not x_2 or vice-versa. I implement the conjunction of entities in a query by combining the *Product of Experts* (Hinton 1999) approach with the *Variational Autoencoder* (VAE) (Kingma and Welling 2014a) method to approximate $p_\theta(z|\mathcal{Q})$.

I first map each x to an approximate posterior $q_\phi(z|x)$ via a neural network $\text{NN}_\phi^{(i)}$ and then I take their product to approximate $p_\theta(z|\mathcal{Q})$.

$$p_\theta(z|\mathcal{Q}) \approx q_\phi(z|\mathcal{Q}) \propto \prod_{x \in \mathcal{Q}} q_\phi(z|x).$$

The ϕ parameters are estimated by minimizing $KL(q(z|x) || p(z|x))$ as shown in § 4.5.3.⁸ The benefit of the POE approximation is that the posterior approximation $q_\phi(\cdot|x)$ for each entity x in \mathcal{Q} acts as an expert and the product of these experts will assign a high value to only that region where all the posteriors assign a high value. Therefore the POE approximation is a way of implementing conjunctive semantics for a query. Another benefit is that if $q_\phi(\cdot|x)$ is an exponential family distribution with a constant base measure whose natural parameters are the output of $\text{NN}_\phi^{(i)}$, then the product of the distributions $\prod_x q_\phi(\cdot|x)$ lies in the same exponential family whose natural parameters are simply the sum of individual neural network outputs. Also, notice that the POE approach recommends adding the *outputs* of the neural networks which is different than

⁸ This is a generalization of (Bouchacourt, Tomioka, and Nowozin 2017) combining variational approximations of posterior distributions since the product of Gaussians is a Gaussian distribution.

concatenating the features for all x in \mathcal{Q} or naively adding the *inputs* of the neural network.⁹

I now show in more detail how the product of experts can be computed simply by adding the output of the neural networks in the special case that the variational approximation has the following form:

$$q_\phi(z|x) \propto h(z) \exp(\langle \psi(z), \text{NN}_\phi^{(i)}(x) \rangle) \quad (4.3)$$

where $\psi(z)$ are the features of z . If h is constant – which is true for many exponential family distributions such as the Bernoulli, Exponential, Pareto, Laplace, Gaussian, Gamma, and the Wishart distributions – then:

$$q_\phi(z|x) \propto \exp(\langle \psi(z), \text{NN}_\phi^{(i)}(x) \rangle).$$

In turn,

$$\prod_{x \in \mathcal{Q}} q_\phi(z|x) \propto \exp(\langle \psi(z), \sum_{x \in \mathcal{Q}} \text{NN}_\phi^{(i)}(x) \rangle).$$

This shows that the product of experts can be computed simply by summing the outputs of the neural network activations for such *deep-exponential* families with constant base measure.

I use $\text{NN}_\phi^{(i)}$ to compute the mean and log-variance of the gaussian distribution $q_\phi(z|x)$ (4.4) that I then convert to the natural parameters of a Gaussian (4.5). Next, I add the natural parameters of the individual variational approximations ξ_x, Γ_x to compute the parameters $\xi_{\mathcal{Q}}, \Gamma_{\mathcal{Q}}$ for $q_\phi(z|\mathcal{Q})$ (4.6). Finally, I compute

⁹Recently, (Zaheer et al. 2017) gave a theorem that any permutation invariant function of sets must be representable as the function of a sum of features of elements of the set. I note that my POE approximation also has a similar form and is permutation invariant.

$q_\phi(z|\mathcal{Q})$ (4.7).

$$\mu_x, \Sigma_x = \text{NN}_\phi^{(i)}(f_x) \quad (4.4)$$

$$\xi_x, \Gamma_x = \mu_x \Sigma_x^{-1}, \Sigma_x^{-1}. \quad (4.5)$$

$$\xi_{\mathcal{Q}}, \Gamma_{\mathcal{Q}} = \sum_{x \in \mathcal{Q}} \xi_x, \sum_{x \in \mathcal{Q}} \Gamma_x. \quad (4.6)$$

$$q_\phi(z|\mathcal{Q}) = \mathcal{N}_c(z|\xi_{\mathcal{Q}}, \Gamma_{\mathcal{Q}}) \quad (4.7)$$

As explained above, the benefit of using the natural parameterization is that I can simply add the natural parameters of the individual variational approximations ξ_x, Γ_x to compute the parameters $\xi_{\mathcal{Q}}, \Gamma_{\mathcal{Q}}$ for $q_\phi(z|\mathcal{Q})$ as

$$\xi_{\mathcal{Q}}, \Gamma_{\mathcal{Q}} = \sum_{x \in \mathcal{Q}} \xi_x, \sum_{x \in \mathcal{Q}} \Gamma_x. \quad (4.8)$$

Finally, I compute $q_\phi(z|\mathcal{Q})$ such that

$$q_\phi(z|\mathcal{Q}) = \mathcal{N}_c(z|\xi_{\mathcal{Q}}, \Gamma_{\mathcal{Q}}),$$

where $\mathcal{N}_c(z|\xi, \Gamma)$ is the multi-variate Gaussian distribution in terms of its natural parameters –

$$\frac{|\Gamma|^{1/2}}{(2\pi)^{D/2}} \exp \left(-\frac{(z^T \Gamma z - 2\xi^T z + \xi^T \Gamma^{-1} \xi)}{2} \right).$$

4.5.2 Inference Step 2: Entity Ranking

In order to rank the entities $x \in \mathcal{R}$, I design a similarity score between the probability distributions $q_\phi(z|\mathcal{Q})$ and $q_\phi(z|x)$ as an efficient substitute for bayesian model comparison. I use the distance between precision weighted means $\xi_{\mathcal{Q}}$ and ξ_x to define my “distance” function as

$$\text{score}_{NVSE}(\mathcal{Q}, x) = -||\xi_{\mathcal{Q}} - \xi_x||^2. \quad (4.9)$$

My inter-distribution “distance” is not a proper distance because it changes as the location of both the input distributions is shifted by the same amount. I experimented with more standard, reparameterization invariant, divergences and kernels such as the KL-divergence and the Probability Product Kernel (Jebara, Kondor, and Howard 2004), see (Appendix A.2), but I found my approach to be faster and more accurate. I believe this is because the regularization from the prior that encourages the posteriors to be close to the origin makes shift invariance unnecessary.

4.5.3 Unsupervised Training

In general VAEs are a combination of deep neural generative models and deep approximations of posterior distributions of such generative models. NVSE is trained in an unsupervised fashion to learn its parameters θ and ϕ . (Kingma and Welling 2014a; Rezende, Mohamed, and Wierstra 2014) proposed the VAE framework for learning richly parameterized conditional distributions $p_\theta(x|z)$ from unlabeled data. I follow (Kingma and Welling 2014a)’s reparameterization trick to train a VAE and maximize the *Evidence Lower Bound*:

$$E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)). \quad (4.10)$$

During training, I do not have access to any clustering information or side information that tells us which entities can be grouped. Therefore I assume that the entities $x \in \mathcal{X}$ were generated i.i.d. The model during training looks the same as Figure 4.2 but with one difference: Q is a singleton set of just one entity.¹⁰ Note that my learning method requires no supervision in contrast to methods

¹⁰More informally, I remove the plates from Figure 4.2.

like Deep Sets which require cluster information, or Neural Collaborative filtering methods which require a large dataset of user interactions.

To learn the parameters during training, I update ϕ and θ using stochastic back-propagation.

4.5.4 Support for weighted queries

Useful recommendation systems for users should be tunable. If a recommendation system returns undesirable entities in response to the query, then the user should be able to easily tune the query so that the system excludes the undesirable results. Most search engines allow boolean exclusion operators or weighted query terms, but in the ESE systems presented so far, a user can only change a query by either removing or adding entities. Furthermore, Weighted-queries enable users to tell the system what aspects of the query to focus on or ignore.

To apply user provided weights as the amount of influence that an entity should have on the final posterior over topics, I integrate the weights directly into the computation of the topic posteriors. If the user provides weights $\tau = \{\tau_x \mid x \in \mathcal{Q}\}$, I compute the query features as

$$\xi_{\mathcal{Q},\tau}, \quad \Gamma_{\mathcal{Q},\tau} = \sum_{x \in \mathcal{Q}} \tau_x \xi_x, \quad \sum_{x \in \mathcal{Q}} |\tau_x| \Gamma_x. \quad (4.11)$$

BM25 supports weights by multiplying each f_x by x 's weights when computing $\bar{f}_{\mathcal{Q}}$. It is not clear how to incorporate weights in Bayesian Sets. instead of computing $\Gamma_{\mathcal{Q}} = \sum_{x \in \mathcal{Q}} \text{NN}_{\phi}^{(i)}(x)$ in (4.6), I perform a weighted sum

$$\Gamma_{(\tau,\mathcal{Q})} = \sum_{x \in \mathcal{Q}} |\tau_x| \text{NN}_{\phi}^{(i)}(x)$$

Note that in computing the precision Γ I only use the magnitude of the provided

weights. To allow a user to tell the system to focus specifically on entities *NOT* similar to a specific x , I enable a user to add negative weights. The signed weights are used for computing ξ as follows:

$$\xi_{(\tau, \mathcal{Q})} = \sum_{x \in \mathcal{Q}} \tau_x \xi_x$$

4.6 Interpretability

I introduce a general approach for interpreting ESE models based on *query rationales* to explain the latent concept the model discovered and *result justifications* to provide evidence for why the system ranked an entity highly.

Useful recommendation systems for users should be tunable. If a recommendation system returns undesirable entities in response to the query, then the user should be able to quickly tune the query so that the system excludes the undesirable results. Most search engines allow boolean exclusion operators or weighted query terms, but in the ESE systems presented so far, a user can only change a query by either removing or adding entities. However, with the NVSE system, based on query rationales and result justifications, users can add weights to entities in a query to tell the system what aspects of the query to focus on or ignore.

4.6.1 Query Rationale

A *Query Rationale* is a visualization of the latent beliefs of the ESE system given the query \mathcal{Q} . Given \mathcal{Q} , I constructed a feature-importance-map $\gamma_{\mathcal{Q}}$ that measures the relative importance of the features in f_x and I show the top features according to $\gamma_{\mathcal{Q}}$ as “Query Rationales”. Recall that the j^{th} component of f_x ,

associated with entity x , measures how often the j^{th} feature co-occurred with x .

I now present how I constructed $\gamma_{\mathcal{Q}}$ for NVSE and the baselines.

For BM25, $\gamma_{\mathcal{Q}}$ is simply $\bar{f}_{\mathcal{Q}}$. In BS, $\gamma_{\mathcal{Q}}$ is the weights from (A.1b): for each j^{th} component of f_x ,

$$\gamma_{\mathcal{Q}}[j] = \log \frac{\tilde{\alpha}_{\mathcal{Q}}[j]\beta[j]}{\alpha[j]\tilde{\beta}_{\mathcal{Q}}[j]}.$$

The benefit of generative methods such as BS and NVSE is that for them query rationales can be computed as a natural by-product of the generative process instead of as ad-hoc post-processing steps. For NVSE, ideally $\gamma_{\mathcal{Q}}$ should be the posterior distribution $p_{\theta}(f|\mathcal{Q})$. Since this is intractable I approximate it by sampling the inference network:

$$p_{\theta}(f|\mathcal{Q}) = E_{p_{\theta}(z|\mathcal{Q})}[p_{\theta}(f|z, \mathcal{Q})] \approx E_{q_{\phi}(z|\mathcal{Q})}[p_{\theta}(f|z)].$$

I further approximate the expectation with a single sample of the mean of $q_{\phi}(z|\mathcal{Q})$. Finally the feature importance map for NVSE is:

$$\gamma_{\mathcal{Q}} = p_{\theta}(f|E[q_{\phi}(z|\mathcal{Q})]).$$

Because Word2Vecf finds the nearest-neighbor between entity embeddings, which are produced through a complicated learning process operating on the whole text corpus, it does not provide a natural way to determine the importance of a single sentence and therefore it is not possible to say what was the effect of a particular sentence on the query results. Similarly, since the SetExpan method works by extracting context features and iteratively expanding this feature set, it is not possible to determine the effect of a single sentence on the final search results.

4.6.2 Result Justifications

I define result justifications as sentences in \mathcal{M}_x that justify why an entity was ranked highly for a given query. Ranking the sentences that mention an entity is similar to ranking entities in \mathcal{R} . Just as I create a feature vector for each x , I create a feature vector for each sentence in \mathcal{M}_x and use the same scoring function to rank the sentences based on the query. While computing a score for entity x based on a query, I also score each sentence in \mathcal{M}_x . My approach to generating interpretable result justifications is agnostic to ESE methods with the caveat that for methods like Word2Vecf and SetExpan this will require retraining or reindexing over the corpus for each query. My approach will not be feasible for such methods.

4.6.3 Weighted queries

Any recommendation system can occasionally fail to provide good results for a query. To improve a system’s responses in such cases, I enable users to guide NVSE’s results by using entity weights to influence the posterior distribution over topics.

If a user provides weights $\boldsymbol{\tau} = \{\tau_x \mid x \in \mathcal{Q}\}$, I compute the query features via Eq. 4.11. The above formulae have an intuitive explanation that when an entity has a higher weight, then the precision over the concepts activated by that entity is increased according to the magnitude of the weight, and the value of the precision weighted mean is also weighted by the user-supplied weights. In turn, an entity with zero weight has zero effect on the final search result and entities with a high negative weight return entities diametrically opposite to

that entity with higher confidence.

Weights can be applied to other methods as well. BM25 can multiply each f_x by x 's weights when computing \bar{f}_Q , and Word2Vecf can use a weighted average. It is not straight-forward to incorporate weights in BS and SetExpan systems. One possible way is to use bootstrap resampling of the query entities according to a softmax distribution over entity weights, but bootstrapping makes the system non-deterministic and therefore even more opaque for a user. Also, bootstrap resampling requires multiple query executions, and it is not straight-forward to combine the outputs of different search queries; therefore I do not advocate for bootstrapping.

4.7 Comparative Experiments

My proposed method determines the latent random variable responsible for generating the query and then ranks the entities in \mathcal{R} by computing a distance between the probability of the latent variable given the given query and the probability of the latent variable given each entity. I test the hypothesis that NVSE can help bridge the gap between advances in IR and real-world use cases. I use human annotators on Amazon Mechanical Turk (AMT) to determine whether NVSE finds more relevant entities than my baseline methods in a real world, automatically generated KG.

4.7.1 Dataset

TinkerBell (Al-Badrashiny et al. 2017) is a KG construction system that achieved top performance in TAC-KGP2017 evaluation.¹¹ I used it as my automatic KG. For each entity e in TinkerBell I create \mathcal{M}_e by concatenating all sentences that mention e and remove the top 100 most frequent features in the corpus from \mathcal{M}_e to clean stop words. Tinkerbell was constructed from the TAC KGP 2017 evaluation source corpus, LDC2017E25, that contains 30K English documents and 60K Spanish and Chinese documents.¹² Half of the English documents come from online discussion forums and the other half from news sources, e.g., Reuters or the New York Times. My experiments only use the 77,845 EDL entities within TinkerBell that are assigned the type **Person**. I use these links to create a map from DBpedia categories to entities in TinkerBell, say M . Each entity in TinkerBell is associated with spans of characters that mention that entity. I tokenize and sentence segment the documents in LDC2017E25 and associate sentences to each entity corresponding to mentions. In the end, I get 344,735 sentences associated with the 77K entities. The median number of sentences associated with an entity is 1, and the maximum number of sentences is 4638 for the *Barack Obama* entity.¹³ This is a good example of how automatic KGs differ from manually curated KGs. In TinkerBell most of the entities appear in only a single sentence so only a single fact may be known about them. In contrast,

¹¹Tinkerbell constructed a KG from LDC2017E25 that contains 30K English documents. Half of them are from online forums and the other half from Reuters and NYT. I focused on the 77,845 entities from English documents appearing in 344,735 sentences. 25,149 entities were also linked to DBpedia.

¹²tac.nist.gov/2017/KGP/data.html

¹³The Mean is 4.43, the standard deviation is 29.19, the minimum number of sentences for an entity is 1, the maximum number of sentences is 4638, and the median is 1 (44,317 entities).

KGs like FreeBase and DBpedia have a more uniform coverage of facts for entities present in them. Another difference is that relational information such as ancestry relations between entities are noisier in an automatically generated KB than in DBpedia which relies on manually curated information present in Wikipedia.

4.7.2 Implementation Details

I prune the vocabulary by removing any tokens that occur less than 5 times across all entities. I end up with, $F=105448$, $V = 61311$, $D = 24661$, and $V' = 19476$. I used BM25 implemented in Gensim (Řehůřek and Sojka 2010) and I implemented BS myself. I choose $\lambda = 0.5$, out of 0, 0.5, or 1, after visual inspection. I used Word2Vecf and SetExpan codebases released by the authors.¹⁴ For NVSE, I set $d=50$, $\sigma=1$. The generative network $NN_{\theta}^{(g)}$ does not have hidden layers and the inference network $NN_{\phi}^{(i)}$ has 1 hidden layer of size 500 with a tanh non-linearity and two output layers for the mean μ_x and log of the diagonal of the variance Σ_x . I use a diagonal Σ_x .¹⁵ For Word2Vecf, I used $d = 100$ to use the same number of parameters per entity as in NVSE. I trained with default hyperparameters for 100 iterations. I used SetExpan with the default hyperparameters as well except that I limited the number of maximum iterations to 3 since I only needed top 4 entities for my experiments.

¹⁴<https://bitbucket.org/yoavgo/word2vecf>, github.com/mickeystroller/SetExpan

¹⁵Training NVSE on 1 Tesla K80 using the Adam optimizer with learning rate $5e^{-5}$ and minibatch size 64 took 12 hours.

4.7.3 Experimental Design

Prior work typically evaluates ESE on a small number of queries, constituting the most frequent entities, e.g., (Ghahramani and Heller 2006) reported results for 10 queries with highly cited authors and (Shen et al. 2017) used 20 test queries created of 2000 most frequent entities in Wikipedia. However, in automatic KGs, most entities are mentioned only a few times. For example, 60% of the entities in TinkerBell are mentioned once. I am primarily interested in unbiased evaluation over such entities; therefore I stratified the evaluation queries into three types.

The 1st type contains entities mentioned in only 1 sentence, the 2nd contains entities appearing in 2 – 10 sentences, and the 3rd contains entities mentioned in 11 – 100 sentences. I also stratified queries based on whether they had 3, or 5 entities. For each query type, I randomly generate 80 queries by first sampling 80 Wikipedia categories and then sampling entities from those categories that were also part of the TinkerBell KG. This results in 480 queries. See Table 4.1 for examples.

For each query, I showed the names and first paragraphs from the Wikipedia abstracts of the query’s entities, to help the AMT workers disambiguate entities unfamiliar to them. Then I showed the workers the top 4 entities returned by each system. Each resultant entity was shown with up to 3 *justification* sentences. Figure 4.3 illustrates the AMT interface.

Since SetExpan and Word2Vecf do not return justifications, I used NVSE to extract justifications for their results. I asked workers to rank the systems between 1, the best system, to 3, the worst; and I allowed for ties. The

Search #717

Query Entity	Description
"Iker Casillas"	"Iker Casillas Fernández (born 20 May 1981) is a Spanish football goalkeeper who plays for and captains both La Liga club Real Madrid and the Spanish national team. In 2008 he was the captain of the Spanish team that won their first European Championship in 44 years, the Spanish team that went on to win Spain's first World Cup (a tournament in which he won the Yashin Award) and the 2012 European Championship."
"Sergio Busquets"	"Sergio Busquets Burgos is a Spanish professional footballer who plays for FC Barcelona and the Spanish national team, as a defensive midfielder. He was a relatively obscure player when he arrived in FC Barcelona's first team in July 2008, but eventually made a name for himself in a relatively short period of time, reaching the Spanish national team in less than one year after making his professional club debut."
"Carles Puyol"	"Carles Puyol i Sforçada (born 13 April 1978) is a Spanish professional footballer who plays for FC Barcelona and the Spanish national team. Mainly a central defender he can also play on either flank, especially as a right back."

Results

System 1	System 2	System 3
"Xavi Hernandez" it was all change for barca at home to sevilla as well with players such as xavi hernandez sergio busquets pedro rodriguez jordi alba carles puyol	"Jordi Alba" should either of those two get injured or suspended before puyol is fit youngster marc bartra who clearly doesn't enjoy martinós	"Barca" real madrid coach jose mourinho insisted that his side have no chance of lifting the title as they are 15 points behind barca
"Barca" it was all change for barca at home to sevilla as well with players such as xavi hernandez sergio busquets pedro rodriguez jordi alba carles puyol	"Xavi Hernandez" meanwhile xavi hernandez will drag his vulnerable hamstring back into action and start in midfield alongside sergio busquets	"Barack Obama" the obama dogma of thought haters and haters of free speech demand illegal aliens to flood the usa to take americans and legal alien's jobs
"Leo Messi" messi had put barca 2-0 up by the break david villa put them ahead 10 minutes in the second half and jordi alba sealed the win by scoring on the break in the last minute of injury time	"Iniesta" meanwhile xavi hernandez will drag his vulnerable hamstring back into action and start in midfield alongside sergio busquets and andres iniesta	"Leo Messi" messi rattled the espanyol crossbar with a 30 yard free kick but there were no goals and barca will take their unbeaten record to malaga next week
"Iniesta" both barca and real madrid are likely to make wholesale changes for their weekend games with leo messi who looked a long way off full fitness on tuesday night given the chance to recover while others such as andres iniesta xavi hernandez and sergio busquets make way for players such as thiago alcantara david villa alex song and christian tello	"Gerard Pique" with central defender carles puyol still on his way back to fitness following a long term knee injury alba's absence is a real problem for coach tató martino given that adriano appears to currently be the first choice to cover for habitual central defensive pairing of gerard pique and javier mascherano	"Donald Trump" all of my blue collar friends are already voting for true conservative ✓ 1 - best 2 3 - worst

Figure 4.3: Example of task shown to a crowd-source worker on Amazon Mechanical Turk.

Category	Entities
(1 Sent./Ent.) American Jazz Singers	Paula West, Natalie Cole, Chaka Khan
(2-10 Sent.) Australian Major Golfers	Marc Leishman, David Graham, James Nitties
(11-100 Sent.) The Apprentice (U.S) Contestants	Maria, Rod Blagojevich, Dennis Rodman, Joan Rivers, Piers Morgan

Table 4.1: Examples of randomly created queries

Ents. In Query	Sents. Per Ent.	Group 1			Group 2		
		NVSE	BM25	BS	NVSE	SetEx	W2Vecf
3	1	27	38	15	51	14	15
	2-10	29	25	26	49	13	18
	11-100	35	23	22	44	10	26
5	1	38	25	17	58	19	3
	2-10	40	27	13	53	19	8
	11-100	24	33	24	52	11	17
	Total	193	171	117	307	86	87

Table 4.2: The number of times a system was ranked 1st over 80 queries compared to other systems in the same group. Ties were allowed so some rows may not sum to 80. Bold highlights the system with the most 1st in its group. Extended results with second and third place rankings of the system are shown in Table 4.3.

annotators found it difficult to compare results from 5 systems at a time, so I split my evaluation into two groups. Group 1 compared NVSE to BS and BM25, and group 2 compared NVSE to SetExpan and Word2Vecf. I randomized the placement of the lists so that the workers could not figure out which system created which list.

4.7.4 Results

Table 4.2 shows the number of times the annotators ranked each system as the best out of the 80 queries. Over all queries, NVSE returned better results compared to the 4 baselines systems. It performed best with 5 entities in the

query where each entity was only mentioned up to 10 times in the corpus. This shows that NVSE can discern better quality topics from multiple entities with sparse data. Extended results showing second and third place rankings of the systems are given in Table 4.3 which show that in cases that when NVSE does not rank first, it is typically chosen as the second-ranking system.

Table 4.3 shows the second and third place rankings of the systems and extends the results shown in Table 4.2.

Ents. In Query	Sents. Per Ent.	Group 1			Group 2			Group 1			Group 2		
		NVSE	BM25	BS	NVSE	SetEx	W2Vecf	NVSE	BM25	BS	NVSE	SetEx	W2Vecf
3	1	36	28	16	20	21	39	17	14	49	9	45	26
	2-10	22	36	22	26	22	32	29	19	32	5	45	30
	11-100	24	26	30	23	22	34	21	31	28	12	48	20
5	1	28	37	15	20	47	13	14	18	48	2	14	64
	2-10	22	27	31	21	50	9	18	26	36	6	10	63
	11-100	20	27	32	17	29	34	36	20	24	11	40	29

Table 4.3: The number of times a system was ranked 2^{nd} (left subtable) and 3^{rd} (right subtable) over 80 queries.

The IR method BM25 was the strongest baseline, outperforming BS and SetExpan, and even NVSE in two settings. I believe that this is because of the low-resource conditions of my evaluation where ad-hoc IR methods can have an advantage. Another reason why BM25 worked very well in my evaluation was the lack of auxiliary signals such as entity inter-relations and entity links and because all the entities were of type *person*. This makes my task different from the entity list completion (ELC) task (BALOG 2009) and a bit simpler for methods that focus heavily on lexical overlap. Another difference between the ESE task and the ELC task was that in the ELC task a descriptive prompt describing the query was also given to the users while evaluating the relevance of

the returned results whereas no such prompt was given in the ESE task. I also found that sometimes BM25 was rated highly because it returned results that were highly relevant to a single query entity instead of being topically similar to all entities. For example, on the query associated with “The Apprentice Contestants” BM25’s results solely focused on Dennis Rodman, but NVSE tried to infer a common topic amongst entities and returned generic celebrities which annotators did not prefer.

On entities with little data, Word2Vecf and SetExpan perform poorly. Word2Vecf requires large amounts of data for learning useful representations (Altszyler, Sigman, and Slezak 2016) which explains why it performs poorly in my evaluation. The SetExpan algorithm directly uses context features extracted from the mentions of an entity and returns entities with the same context features. This approach can overfit with low data. Even though SetExpan uses an ensembling method to reduce the variance of the algorithm, I believe using context-features causes overfitting when an entity appears in only a few sentences. Lastly, I believe that BS suffers because its impoverished generative model has neither non-linearities nor low-dimensional topics for modeling correlations amongst tokens.

4.8 Analyzing Interpretability

I now attempt to understand the similarity relations encoded in NVSE’s internal concept representations to understand what it is learning. I also provide examples of how query rationales and query weights can help users fine-tune their queries.

column 3	column 14	column 20	column 33	column 37
merger procurement industry securities premiers AP-doc1 NYT-doc2 analyst protection founders	husband iii sister house <u>she</u> labor <u>her</u> king daughter church	best very its most good end some do only such	game tackle drill fuzzy offensive 21 coach doc artur doc3	wild lighting holly costumes exhibit fashion martin’s nightclub thriller theatrical
<i>business finance</i>	<i>family royalty</i>	<i>qualifier words</i>	<i>football sports</i>	<i>entertainment movie</i>

Table 4.4: The first row contains top 10 features most similar to z_j . The bottom row contains labels agreed upon by the authors; I loosely refer to the group where $j = 20$ as “qualifiers”. Underscored words signify that the feature came from \mathcal{V}' .

Abu Bakr Baghdadi (1)	Osama Bin Laden (1)	O.B. Laden (1.5) A.B. Baghdadi (1)	O.B. Laden (0.5) A.B. Baghdadi (2)	O.B. Laden (-0.2) A.B. Baghdadi (1)
qaida, iraq, abu, baghdadi, bahr, al, leader, attacks	bin, laden, osama, al, cia, pakistani, afri, qaida	qaida, al, u, pak- istani, cia, qaeda, government, killed	qaida, al, leader, attacks, u, killed, officials, islamic	bahr, baghdadi, abu, iraq, al, sec- tarian, nuri, kur- dish

Table 4.5: The top row represents a query with weights in parentheses and the bottom row lists corresponding query rationales.

4.8.1 Understanding the concept space

To gain some insight into the distribution over concepts inferred by NVSE I determined the top 10 words activated by individual dimension of z by computing $\text{NN}_{\theta}^{(g)}(e_j)$ where e_j is a one-hot vector in \mathbb{R}^{50} . Table 4.4 shows the top 10 words for selected components of z . I can easily recognize that dimensions 3, 33 and 37 of z represent finance, sports, and entertainment. Even though I did not constrain z to be component-wise interpretable, this structure naturally emerged after training.

4.8.2 Weights & Query Rationale

Table 4.5 depicts how the *query rationale* returned by NVSE changes in response to entity weights. In the first column the query is {Abu Bakr Baghdadi} and the query rationale tells us that NVSE focuses on *iraq*, *baghdadi* etc. The second column shows a different query {Osama Bin Laden} and the query rationales changes accordingly to *pakistani* and *osama*. The third and fourth column show rationales when the weights on “Laden” and “Baghdadi” are varied. When more weight is put on “Laden” then the query rationales contain more features that are associated to him, and when more weight is put on “Baghdadi”, then features such as “islamic” which is a token from his organization are returned. The last column shows an interesting configuration where a user is effectively asking for results that are similar to “Baghdadi” but dissimilar to “Laden” and the feature for *kurdish* gets activated. Since the system returns results in under 100ms, the user can fine-tune her query in real-time with the help of these query rationales.

I give one more example of the utility of negative weights: When $\mathcal{Q} = \{\text{Brady}\}$, NVSE’s rationale is [*brady*, *game*, *patriots*, *left*, *knee*, *field*, *tackle*], indicating that NVSE associated the “Brady” entity with Tom Brady who is a member of the Patriots football team. When I added “Wes Welker” to \mathcal{Q} with a negative weight, the query rationale changed to [*brady*, *game*, *left*, *tackle*, *knee*, *back*, *field*]. Since Wes is a Patriots receiver who received a negative weight in the query, NVSE deactivated the *patriots* feature and activated the *tackle* feature, opposite to a *receiver*.

4.9 Conclusion

This chapter is based on the following journal publication:

Rastogi, Pushpendre, Adam Poliak, Vince Lyzinski, and Benjamin Van Durme (2018). “Neural variational entity set expansion for automatically populated knowledge graphs”. In: Information Retrieval Journal.

The main ideas and scientific contributions are:

- The first to learn Deep Representation of entities grounded in natural text for the task of Set Expansion.
- The first to propose an efficient way of combining the posterior outputs of a VAE’s inference network from multiple observations by summing the natural parameters.
- One of the first methods for extracting query rationales for entity recommendations given a set expansion query.

I introduced NVSE as a step towards making advances in entity set expansion useful to real-world settings. NVSE is a novel unsupervised approach based on the VAE framework that discovers related entities from noisy knowledge graphs. NVSE ranks entities in a KG using an efficient and fast scoring function (4.9), ranking 80K entities on a commodity laptop in 100 milliseconds.

My experiments demonstrated that NVSE could be applied in real-world settings where automatically generated KGs are noisy. NVSE outperformed state of the art ESE systems and other strong baselines on a real-world KG. I

also presented a flexible approach to interpret ESE methods and justify their recommendations.

In future work, I will extend my work by improving my model using more powerful auto-encoders such as the Ladder VAE (Sønderby et al. 2016); secondly I will experiment with the use of side information such as links from a KG through the use of Graph Convolutional Networks (Kipf and Welling 2017). Third, I will like to quantitatively measure how query rationales and justifications help users in accomplishing their search task. Finally, I will incorporate confidence scores from the KG in my model. Although there may be future work to improve my ESE method, I believe that NVSE serves as a significant step towards utilizing KGs and semantics for information retrieval and understanding in real-world settings.

Chapter 5

Knowledge Base Embeddings under Logical Constraints

Knowledge bases are large repositories of information about the entities in the real world and the relations between them. They can be thought of as large graphs marking the relations between real-world entities as the edges between its vertices. In the previous chapters, I presented algorithms for learning representation of words and entities from unlabeled, unstructured textual corpora. In this chapter, I shift focus from embedding the components of unstructured text to representing the structured information present in knowledge bases. To that end, I follow a two-pronged approach.¹

First, I scrutinize an existing method for embedding knowledge bases and demonstrate its shortcomings in accurately representing asymmetric-transitive relations both theoretically and empirically. I study the effect of the transitivity of a relation on the performance of the RESCAL algorithm by (Nickel, Tresp, and Kriegel 2011), and I demonstrate via a theorem and empirical results that RESCAL is inappropriate for representing transitive-asymmetric relations in a

¹Previous versions of this work were published in (Rastogi, Poliak, and Van Durme 2017) and (Rastogi and Van Durme 2017)

KB.

Second, I present new objectives and training algorithms for encouraging *logical consistency* in the predictions by a knowledge base completion algorithm by incorporating logical constraints into the learning of entity and relation representations during the training of a Knowledge Base Completion (KBC) system. Enforcing logical consistency in the predictions of a KBC system guarantees that the predictions comply with logical rules such as symmetry, implication and generalized transitivity. My method encodes logical rules about entities and relations as convex constraints in the embedding space to enforce the condition that the score of a logically entailed fact must never be less than the minimum score of an antecedent fact. Such constraints provide a weak guarantee that the predictions made by a KBC model will match the output of a logical knowledge base for many types of logical inferences.

5.1 Introduction

Large-scale and highly accurate knowledge bases (KB) such as Freebase and YAGO2 (Hoffart et al. 2013) have been recognized as essential for high performance on natural language processing tasks such as Relation extraction (Dalton, Dietz, and Allan 2014), Question Answering (Yao and Van Durme 2014), and Entity Recognition in informal domains (Ritter et al. 2011). Because of this importance of large scale KBs and since the recall of even Freebase, one of the largest open source KBs, is low² a large number of researchers have presented models for knowledge base completion (KBC). Knowledge Base Completion

²It was reported by Dong et al. in October 2013, that 71% of people in Freebase had no known place of birth and that 75% had no known nationality.

(KBC), or link prediction, is the task of inferring missing edges in an existing knowledge graph.

A popular strategy for KBC is to *embed* the entities and relations in low dimensional continuous vector spaces and to then use the learned *embeddings* for link prediction. In other words, continuous real-valued vectors and matrices are automatically learned that can represent the entities and edges in a knowledge base, and at the time of inference, these real-valued representations are used to predict whether a particular edge exists between two entities. This general strategy can be implemented in many different ways, and I refer the reader to the survey by (Nickel et al. 2016) for more details. Even though the strategy of embedding the elements of a graph is popular for knowledge base completion, theoretical studies of such methods are scarce. More specifically, although many methods have been evaluated empirically on select datasets for KBC, much less attention has been paid to understanding the relationship between the logical properties encoded by a given KB and the KBC method being evaluated.

In this chapter, I demonstrate *theoretically*, and *experimentally*, the adverse effect that asymmetric, transitive relations can have on a KBC method that relies on a single vector embedding of a KB entity. Transitive-asymmetric relations such as the **type of** relation in Freebase (Bollacker et al. 2008) and, the *hyponym* relation in WordNet (Miller 1995) are ubiquitous in KBs and therefore very important (Guha 2015). For my theoretical result, I analyze a widely cited KBC algorithm called RESCAL (Nickel, Tresp, and Kriegel 2011; Toutanova et al. 2015) and I prove that on large KBs that contain a large proportion of asymmetric, transitive relations, methods such as RESCAL will wrongly predict the existence of edges that are the reverse of edges in the training data. I also

present a way to mitigate this problem, by using role sensitive embeddings for entities and I empirically verify that my proposed solution improves performance. Through my experiments, I also discover a drawback in the prevalent evaluation methodology, of randomly sampling unseen edges, for testing KBC models and show that random sampling can overlook errors on special types of edges.

A number of state of the art methods for Knowledge Base Completion (KBC) utilize a representation learning framework and learn distributed vector representations, i.e., *embeddings*, of the entities and relations in a Knowledge Base (KB). Although such models make correct predictions on a sizable portion of the data, they cannot guarantee to follow logical rules and to make inferences consistent with those rules. For example, there is no way to guarantee in existing KBC methods that if an embeddings based KB predicts the fact that *Alice murdered Bob* (`Murdered, (Alice, Bob)`) then it will also predict that *Alice Killed Bob*, even though it is very simple to enforce this in a traditional logical inference system by specifying the rule that `Murdered` implies `Killed`. Consider another example, if a KB knows that `AliceIsAWoman` and that `BobIsSonOf Alice`, but the KBC method cannot guarantee to infer that `AliceIsMotherOf Bob` then such a method will have limited use in real applications.

In the second half of this chapter, I present a novel method for directly encoding logical rules via convex constraints on the embeddings. Such methods for directly “shaping” the feasible subspaces of embeddings based on logical properties of relations have not been deeply explored before, and I will show through my experiments that such a method can improve the performance of an existing KBC system. I validate my method via experiments on a knowledge graph derived from WordNet.

5.2 Related Work

Due to the large body of work that has been done for the task of KBC, it is not possible to cover all of the related work on KBC in this section. Instead, I refer the reader to the survey (Nickel et al. 2016) for an overview of the empirical work that has been done in the area of KBC and link prediction. Similarly, The problem of enforcing consistency between the predictions made by a machine learning system and a first-order logic system, which is what my work attempts to do, has a large history of research but I will only be able to review recent work on learning representations of entities and relations of a knowledge graph and refer the reader to reviews of neural-symbolic systems (Garcez, Gabbay, and Broda 2002; Hammer and Hitzler 2007) for more references.

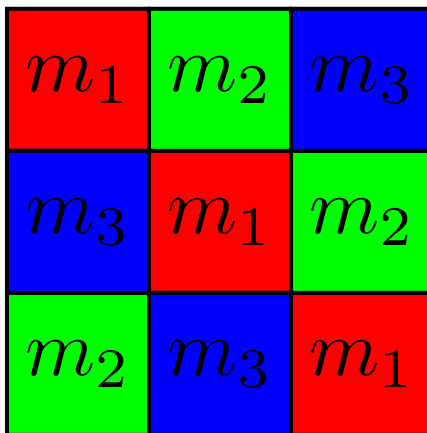
5.2.1 Methods for Knowledge Base Completion

Since I focus on the analysis of RESCAL, my work is most closely related to the paper (Nickel, Jiang, and Tresp 2014). This chapter proves an important theorem that shows that the dimensionality required by the RESCAL model³ for exactly representing a weighted adjacency matrix of a knowledge graph must be greater than the number of strongly connected components in the graph. In my setting where I consider data sets that contain only transitive-asymmetric relations, the number of strongly connected components in the graph equal the number of vertices in the graph. Therefore their theorem proves that the dimensionality required for exactly representing a dataset such as WordNet using an algorithm such as RESCAL must be greater than the number of entities in

³Actually their theorem provides a lower bound for a more general model than RESCAL which automatically applies to RESCAL.

the knowledge graph. In contrast to this result, my analysis gives an explicit example of a type of query for which the RESCAL algorithm will make wrong inferences.

My analysis trivially extends to a few other factorization based algorithms, e.g. the Holographic embedding algorithm by (Nickel, Rosasco, and Poggio 2016). The holographic embedding method can be rewritten as a constrained form of RESCAL with a “holographically constrained” matrix M . Figure 5.1 shows an example of a 3×3 holographically constrained matrix with the constraint that elements with the same color must hold the same value. Since such a matrix is asymmetric by construction, my theorem proves that there will exist vectors a, b , and c for which M will violate transitivity.



m_1	m_2	m_3
m_3	m_1	m_2
m_2	m_3	m_1

Figure 5.1: An illustration of a “holographically constrained” matrix.

Recently (Bouchard, Singh, and Trouillon 2015) argued that the phenomenon of transitivity of relations between vertices in a knowledge graph could be modeled with high accuracy if the knowledge graph is modeled as a thresholded version of a latent low-rank real matrix, and the vertex embeddings are learned as a low-rank factorization of that latent matrix. Based on this argument

they claimed that factorizing a knowledge graph with a squared loss was less appropriate in comparison to factorizing it with a hinge loss or logistic loss. In this work, I provide an argument based on the symmetry of transitive matrices to show that the method of RESCAL which minimizes the squared reconstruction error must fail to capture phenomenon like transitivity in large knowledge bases. In this way, my results complement the work by Bouchard, Singh, and Trouillon.

5.2.1.1 Logically Constrained Representations for KBC

(Grefenstette 2013) presented a novel model for simulating propositional logic with the help of tensors; however, their model relied on high-dimensional boolean embeddings of the entities and relations, and it only guaranteed adherence to the RELIMP rule out of the ones presented in this chapter. (Rocktäschel et al. 2014; Rocktäschel, Singh, and Riedel 2015) generalized Grefenstette’s work learning embeddings of entities and relations that were real-valued and low dimensional and their learning mechanism could accommodate arbitrary first-order logic formulae into the parameter learning objective by propositionalizing the formulae. Their method has two drawbacks in comparison to my proposal — 1. The process of propositionalization can be very expensive, especially for rules like PROTRANS and TYPEIMP that quantify over tuples of entities, and 2. Their method of *differentiation through logic* does not guarantee that the learned embeddings will always be able to predict unseen relations that are logically entailed given the rules and the training data.

(Bowman, Potts, and Manning 2015; Bowman and Potts 2015) presented a neural network-based method for predicting the existence of natural logic relations between two entities. Their approach too had the drawback that it

could not guarantee the inference of logically entailed facts.

(Wang, Wang, and Guo 2015) presented an interesting approach for “batch-mode” knowledge base completion. They proposed to perform KBC in two steps – First, they learned a distributional model of the entities and relations in a KB to predict the likelihood of unobserved facts. In the second step, they optimized a global objective with logical constraints using an ILP solver. Their approach is very different from ours since I present an online method for performing knowledge base completion and I directly translate the logical rules into constraints on the parameters instead of relying on a black box ILP solver.

(Guo et al. 2015) presented a method based on LLE (Roweis and Saul 2000) for incorporating side information in the form of semantic categories of entities, but their method is not capable of incorporating the range of logical rules that I can. (Demeester, Rocktäschel, and Riedel 2016) and (Vendrov et al. 2016) proposed an approach to constrain the learnt embeddings in a way that is identical to the method prescribed by us in Subsection 5.4.1. Our work generalizes their approach in two ways — Firstly, I generalize their proposed constraints by using the language of convex geometry, and secondly, I propose constraints for many more logical rules and score functions than either of the previous two papers.

(Hu et al. 2016) presented an adversarial setup with a *teacher* neural network co-operating with a *student* neural network to regularize the predictions of the student network to follow logical rules; however, their method amounts to propositionalization of the logical rules. Their method is more general than ours since it can be used to train neural networks however again it lacks guarantees during inference. (Wang and Cohen 2016) presented a novel method of factorizing

the adjacency matrix of a proof graph of a probabilistic logic language to learn embeddings of first-order logic formulas. My method is conceptually simpler than theirs and requires fewer training stages. Finally, (Guo et al. 2016) proposed an alternative method for embedding rules and entities based on t-norm fuzzy logics which was very similar to (Rocktäschel, Singh, and Riedel 2015)’s approach.

5.3 Theoretical Analysis of RESCAL

Notation: A KB contains $(subject, relation, object)$ triples. Each triple encodes the fact that a *subject* entity is related to an *object* through a particular *relation*. Let \mathcal{V} and \mathcal{R} denote the set of entities and relationships. I use \mathcal{V} to denote entities to evoke the notion that an entity corresponds to a vertex in the knowledge graph. I assume that \mathcal{R} includes a type for the *null relation* or *no relation*. Let $V = |\mathcal{V}|$ and $R = |\mathcal{R}|$ denote the number of entities and relations. I use v and r to denote a generic entity and relation respectively. The shorthand $[n]$ denotes $\{x | 1 \leq x \leq n, x \in \mathbb{N}\}$. Let \mathcal{E} denote the entire collection of facts and let e denote a generic element of \mathcal{E} . Each instance of e is an edge in the knowledge graph. I refer to the subject, object and relation of e as $e^{sub}, e^{obj} \in \mathcal{V}$ and $e^{rel} \in \mathcal{R}$ respectively. $E = |\mathcal{E}|$ is the number of known triples.

RESCAL: The RESCAL model associates each entity v with the vector $a_v \in \mathbb{R}^d$ and it represents the relation r through the matrix $M_r \in \mathbb{R}^{d \times d}$. Let v and v' denote two entities whose relationship is unknown, and let $s(v, r, v') = a_v^T M_r a_{v'}$, then the RESCAL model predicts the relation between v and v' to be: $\hat{r} = \operatorname{argmax}_{r \in \mathcal{R}} s(v, r, v')$. Note that in general if the matrix M_r is asymmetric then the score function s would also be asymmetric, i.e., $s(v, r, v') \neq s(v', r, v)$.

Let $\Theta = \{a_v | v \in \mathcal{V}\} \cup \{M_r | r \in \mathcal{R}\}$.

Transitive Relations and RESCAL: In addition to relational information about the binary connections between entities, many KBs contain information about the relations themselves. For example, consider the toy knowledge base depicted in Figure 5.2. Based on the information that **Fluffy** *is-a* **Dog** and that a **Dog** *is-a* **Animal** and that *is-a* is a transitive relations I can infer missing relations such as **Fluffy** *is-a* **Animal**.

Let us now analyze what happens when I encode a transitive, asymmetric relation. Consider the situation where the set \mathcal{R} only contains two relations $\{r_0, r_1\}$. r_1 denotes the presence of the *is-a* relation and r_0 denotes the absence of that relation. The embedding based model can only follow the chain of transitive relations and infer missing edges using existing information in the graph if for all triples of vertices v, v', v'' in \mathcal{V} for which I have observed $(v, \textit{is-a}, v')$ and $(v', \textit{is-a}, v'')$ the following holds true:

$$s(v, r_1, v') > s(v, r_0, v') \text{ and } s(v', r_1, v'') > s(v', r_0, v'') \implies s(v, r_1, v'') > s(v, r_0, v'')$$

$$\text{I.e. } a_v^T(M_{r_1} - M_{r_0})a_{v'} > 0 \text{ and } a_{v'}^T(M_{r_1} - M_{r_0})a_{v''} > 0 \implies a_v^T(M_{r_1} - M_{r_0})a_{v''} > 0 \quad (5.1)$$

I now define a *transitive matrix* and state a theorem that I prove in § 5.3.1.

Definition A matrix $M \in \mathbb{R}^{d \times d}$ is transitive if $a^T M b > 0$ and $b^T M c > 0$ implies $a^T M c > 0$.

Theorem 1. *Every transitive matrix is symmetric.*

If I enforce the constraint in Equation 5.1 to hold for all possible vectors and not just a finite number of vectors then $M_{r_1} - M_{r_0}$ is a transitive matrix.

By Theorem 1, $M_{r_1} - M_{r_0}$ must be symmetric. This further implies that if $s(v, r_1, v') > s(v, r_0, v')$ then $s(v', r_1, v) > s(v', r_0, v)$. In terms of the toy KB shown in Figure 5.2; if the RESCAL model predicts that **Fluffy** *is-a* **Animal** then it will also predict that **Animal** *is-a* **Fluffy**.

Augmenting RESCAL to Encode Transitive Relations: The analysis above points to a simple way for improving RESCAL’s performance on asymmetric, transitive relations. The reason that the original method fails to satisfactorily encode transitive asymmetric relations is because if the score $s(v, r_1, v')$ is high then $s(v', r_1, v)$ will also be high. I can avoid this situation by using two different embeddings for all the entities and compute the score of a relation through those role specific embeddings; i.e. I can use the embeddings a_v^1, a_v^2 to represent vertex v and let $s(v, r_1, v') = a_v^1 M_{r_1} a_{v'}^2$ and $s(v', r_1, v) = a_{v'}^1 M_{r_1} a_v^2$. This idea of using role-specific embeddings has been known for a long time starting from (Tucker 1966).⁴ The specific method that I have just explained is generally known to KBC researchers as the Tucker2 decomposition (Singh, Rocktäschel, and Riedel 2015). To encode more than one relations, only the matrix M_r needs to change, but the entity embeddings can be shared across all relations.

5.3.1 Proof of Theorem 1

I now present my novel proof of Theorem 1 beginning with a lemma.⁵

Lemma 2. *Every transitive matrix is PSD.*

Proof. Consider the triplet of vectors $c := x, b := Mc, a := Mb$. Then $a^T(Mb) =$

⁴Recently (Yoon et al. 2016) used this idea of using role-specific embeddings to preserve the properties of symmetry and transitivity in *translation based* knowledge base embeddings.

⁵Theorem 1 was first proven by (Grinberg 2015)(unpublished). My proof is more elementary and direct.

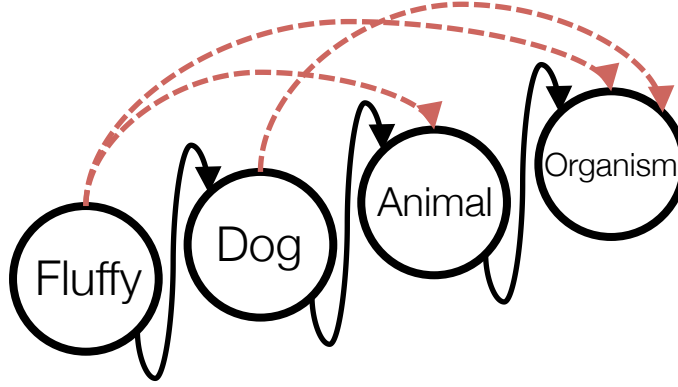


Figure 5.2: A toy knowledge base containing only *is-a* relations. The dashed edges indicate unobserved relations that can be recovered using the observed edges and the fact that *is-a* is a transitive relation.

$\|Mb\|^2 \geq 0$ and $b^T(Mc) = \|b\|^2 \geq 0$ and $a^T Mc = b^T Mb$. Three cases are possible, either $b = 0$, or $Mb = 0$, or both $Mb \neq 0$ and $b \neq 0$. In the third case transitivity applies and I conclude that $b^T Mb > 0$. In all cases $b^T Mb \geq 0$ which implies M is PSD. \square

The next lemma proves that if M is transitive then $x^T My$ and $x^T M^T y$ must have the same sign.

Lemma 3. *If $\exists x, y$ $x^T My > 0$ but $x^T M^T y < 0$ then M is not transitive.*

Proof. Let x, y be two vectors that satisfy $x^T My > 0$ and $x^T M^T y < 0$. Since $x^T M^T y = y^T Mx$ therefore $y^T M(-x) > 0$. If I assume M is transitive, then $x^T M(-x) > 0$ by transitivity, but Lemma 2 shows such an x cannot exist. \square

Lemma 4 is a general statement about all matrices which states that if the two bilinear forms have the same sign for all inputs then they have to be scalar multiples of each other. I omit its proof due to space constraint.

Lemma 4. *Let $M_1, M_2 \in \mathbb{R}^{d \times d} \setminus \{0\}$. If $x^T M_1 y > 0 \implies x^T M_2 y > 0$ then $M_1 = \lambda M_2$ for some $\lambda > 0$.*

Finally I use Lemma 3–4 to prove Theorem 1.

Proof. Let M be a transitive matrix and let x, y be two vectors such that $x^T M y > 0$. By transitivity of M and Lemma 3 $x^T M^T y > 0$. Therefore by Lemma 4 I get $M = \lambda M^T$ for some $\lambda > 0$. Clearly $\lambda = 1$, this concludes the proof that M is symmetric. \square

5.3.2 Experimental Results

My theoretical result in § 5.3 was derived under the assumption that the constraint 5.1 held over all vectors in \mathbb{R}^d instead of just the finite number of vector triples used to encode the KB triples. It is intuitive that as the number of entities inside a KB increases my assumption will become an increasingly better approximation of reality. Therefore my theory predicts that the performance of the RESCAL model will degrade as the number of entities inside the KB increases and the dimensionality of the embeddings remains constant. I perform experiments to test this prediction of my theory.

5.3.2.1 Experiments On Simulated Data

I tested the applicability of my analysis by the following experiments: I started with a complete, balanced, rooted, directed binary tree \mathcal{T} , with edges directed *from* the root *to* its children. I then augmented \mathcal{T} as follows: For every tuple of distinct vertices v, v' I added a new edge to \mathcal{T} if there already existed a directed path starting at v and ending at v' in \mathcal{T} . I stopped when I could not add any more edges without creating multi-edges. For the rest of the chapter, I denote this resulting set of ordered pairs of vertices as \mathcal{E} and those pairs of vertices

that are not in \mathcal{E} as \mathcal{E}^c . For a tree of depth 11, $V = 2047$, $E = 18,434$ and $|\mathcal{E}^c| = 4,171,775$. See Figure 5.3 for an example of $\mathcal{E}, \mathcal{E}^c$.

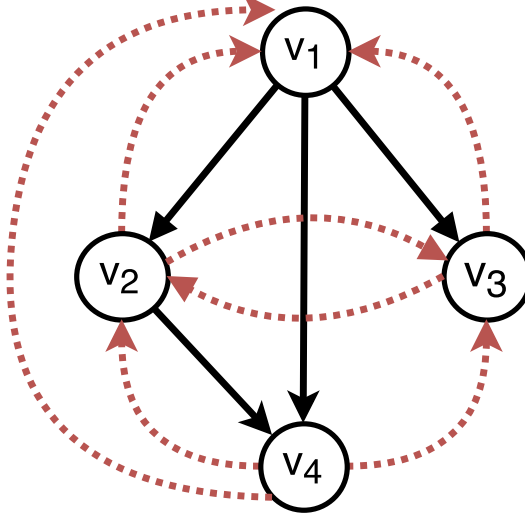


Figure 5.3: Assume that the black edges constitute \mathcal{E} and the red dotted denote \mathcal{E}^c , then \mathcal{E}^{rev} contains the edges (v_4, v_1) , (v_4, v_2) , (v_2, v_1) , and (v_3, v_1) .

I trained the RESCAL model under two settings: In the first setting, called *FullSet*, I used entire \mathcal{E} and \mathcal{E}^c for training. In the second setting, called *SubSet*, I randomly sample \mathcal{E}^c and select only $E = |\mathcal{E}|$ edges from \mathcal{E}^c . All the edges in \mathcal{E} including all the edges in the original tree are always used during both *FullSet* and *SubSet*. For both the settings of *FullSet* and *SubSet* I trained RESCAL 5 times and evaluated the models' predictions on \mathcal{E} , \mathcal{E}^c and $\mathcal{E}^{(rev)}$. $\mathcal{E}, \mathcal{E}^c$ have already been defined, and $\mathcal{E}^{(rev)}$ is the set of reversed ordered pairs in \mathcal{E} . I.e., $\mathcal{E}^{rev} = \{(u, v) | (v, u) \in \mathcal{E}\}$

For every edge in these three subsets, I evaluated the model's performance under 0 – 1 loss. Specifically, to evaluate the performance of RESCAL on an edge $(v, v') \in \mathcal{E}$ I checked whether the model assigns a higher score to (v, r_1, v') than (v, r_0, v') and rewarded the model by 1 point if it made the right prediction and 0

otherwise. As before, r_1 and r_0 denote the presence and absence of relationship respectively.

Note that low performance on \mathcal{E}^{rev} and high performance on \mathcal{E} will indicate exactly the type of failure predicted from my analysis. I vary the dimensionality of the embedding d , and the number of entities V , since they influence the performance of the model, and present the results in Table 5.1a–5.1b. The rightmost column of Table 5.1b is the most direct empirical evidence of my theoretical analysis. The performance of RESCAL embeddings is substantially lower on \mathcal{E}^{rev} in comparison to $\mathcal{E}, \mathcal{E}^c$. The last row with $d = 400$, however, shows a very sharp drop in the accuracy on \mathcal{E}^c while the performance of \mathcal{E}^{rev} increases slightly. I believe that this happens because of higher overfitting to the forward edges as the number of parameters increases.

5.3.2.2 Experiments On WordNet

WordNet is a KB that contains vertices called *synsets* that are arranged in a tree-like hierarchy under the *hyponymy* relation. The hyponym of a synset is another synset that contains elements that have a more specific meaning. For example, the *dog* synset⁶ is a hyponym of the *animal* synset and an *animal* is a hyponym of *living_thing* therefore a *dog* is a hyponym of *living_thing*. I extracted all the hyponyms of the *living_thing.n.01* synset as the vertices of \mathcal{T} and I used the transitive closure of the direct hyponym relationship between two synsets as the edges of \mathcal{T} . Quantitatively, the *living_thing* synset contained 16,255 hyponyms, and 16,489 edges. After performing the transitive closure \mathcal{E}

⁶A synset must be qualified by the word sense and the part of speech. So a valid synset called *dog.n.01*. For simplicity I skip this detail in my explanation but my implementation distinguishes between the synset *dog.n.01* and *dog.n.02*.

<i>FullSet</i>									
d	V = 2047			4095			8191		
	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}
50	66	100	100	60	100	100	54	100	100
100	76	100	100	69	100	100	63	100	100
200	86	100	100	79	100	100	72	100	100
400	94	100	100	88	100	100	81	100	100

(a) Accuracy in percentage of RESCAL with all the edges as training data (denoted as *FullSet*) on $\mathcal{E}, \mathcal{E}^c, \mathcal{E}^{rev}$.

<i>SubSet</i>									
d	V = 2047			4095			8191		
	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}
50	100	93	52	100	91	48	100	89	44
100	100	78	58	100	92	56	100	89	52
200	100	60	72	100	71	61	100	90	59
400	100	54	67	100	57	70	100	65	62

(b) Accuracy in percentage of RESCAL trained with all positive edges and subsampled negative edges as training data (together called *SubSet*).

Table 5.1: V denotes the number of nodes in the tree. d denotes the number of dimensions.

became 128, 241.

I performed two experiments with the WordNet graphs, using the same *FullSet* and *SubSet* protocols described earlier. The results are in the left half of Table 5.2. I see that even though the accuracy on \mathcal{E} and \mathcal{E}^c is high, the performance on \mathcal{E}^{rev} is much lower. This trend is in line with my theoretical prediction that the RESCAL model will fail on “reverse relations” as the KB’s size increases.

d	<i>FullSet</i>			<i>SubSet</i>			<i>SubSet</i>		
	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}	\mathcal{E}	\mathcal{E}^c	\mathcal{E}^{rev}
50	71	100	100	100	93	58	100	55	65
100	79	100	100	100	94	60	100	56	56
200	84	100	100	100	93	63	100	56	75
400	89	100	100	100	68	69	100	97	91

Table 5.2: Results from experiments on WordNet. I used the subtree rooted at the *living_things* synset from the WordNet hierarchy. d indicates the dimensionality of the embeddings used and the triple of numbers under *FullSet* and *SubSet* indicates the accuracy of RESCAL on $\mathcal{E}, \mathcal{E}^c, \mathcal{E}^{rev}$. V is 16, 255 for all columns. The right half shows results from experiments on WordNet with role dependent embeddings for entities.

Finally, I present the results of augmenting RESCAL with role-specific embeddings in the right half of Table 5.2. The results show that using role-specific embeddings increases the performance over the performance of the RESCAL algorithm and with a high dimensionality of embeddings it is possible to encode both the forward and the reverse relations in the embeddings. Please note that I do not claim that my proposed augmentation for RESCAL will empirically be any better than the much more recently proposed methods such as ARE (Nickel, Jiang, and Tresp 2014), or Poincaré embeddings (Nickel and Kiela 2017). I leave a careful empirical comparison of these techniques for future work.

5.3.3 Discussion

The information present in large scale knowledge bases has helped in moving information retrieval beyond retrieval of documents to more specific entities and objects. And in order to further improve coverage of knowledge bases, it is important to research knowledge base completion methods. Since many knowledge bases contain information about real-world artifacts that obey hierarchical relations and logical properties, it is important to keep such properties in mind while designing knowledge base completion algorithms. In this chapter, I demonstrate a close connection between the logical properties of relations such as asymmetry, and transitivity, and the performance of KBC algorithms used to predict those relations. Specifically, I theoretically analyzed a popular KBC algorithm named RESCAL, and my analysis showed that the performance of that model in encoding transitive and asymmetric relations must degrade as the size of the KB increases. My experimental results in Table 5.1a, 5.1b and 5.2 confirmed my theoretical hypothesis, and most strikingly I observed that the accuracy of RESCAL on \mathcal{E}^{rev} was substantially lower than its performance on either \mathcal{E} or \mathcal{E}^c , even though \mathcal{E}^{rev} is a subset of \mathcal{E}^c .

In Table 5.3, I visualize the errors made by RESCAL by listing a few edges in \mathcal{E}^{rev} that were wrongly predicted as true edges. These examples show that the trained RESCAL model can predict that *fruit tree* is a hyponym of *mango* or that every *accountant* is a *bean counter*. Such wrong predictions can be harmful. Based on my analysis, I advocated for role-specific embeddings as a way of alleviating this shortcoming of RESCAL, and I empirically showed its efficacy in Table 5.2.

My results also highlight a problem with the commonly employed KBC evaluation protocol of randomly dividing the edge set of a graph into train and test sets for measuring knowledge base completion accuracy. For example with $d = 50$ the average accuracy on both \mathcal{E} and \mathcal{E}^c is quite high but on \mathcal{E}^{rev} accuracy is low even though \mathcal{E}^{rev} is a subset of \mathcal{E}^c . Such a failure will stay undetected with existing evaluation methods.

<i>Argument 1</i>	<i>Argument 2</i>
draftsman.n.02	cartoonist
fruit tree	mango
taster	wine taster
accountant	bean counter
scholar.n.03	rhodes scholar

Table 5.3: Examples of wrong predictions for the hyponym relations by the RESCAL model with $d = 400$ when trained under the *SubSet* setting. The default synset is *n.01*. i.e. the default synset in this table is the sense 1 for nouns.

5.4 Training Relation Embeddings under Logical Constraints

Let a knowledge base be defined as a tuple $(\mathcal{F}, \mathcal{L})$, with \mathcal{F} a set of statements, and a set of first order logic rules \mathcal{L} . Every element $f \in \mathcal{F}$ is itself a nested tuple $(r, (e, e'))$ which states that the entities e and e' are connected via the relation r . Let \mathcal{E} and \mathcal{R} be the set of all entities and relations respectively. Let \mathcal{T} be the set of all entity tuples that appear in \mathcal{F} , and let \mathcal{U} denote the universe of all possible facts, i.e. $\mathcal{T} = \{t \mid (r, t) \in \mathcal{F}\}$, and $\mathcal{U} = \{(r, (e, e')) \mid r \in \mathcal{R}, e, e' \in \mathcal{E}\}$. Note that $|\mathcal{T}| \leq |\mathcal{U}|$.⁷ Finally, $\mathcal{F}^c = \mathcal{U} \setminus \mathcal{F}$ is the set of unknown facts. The goal

⁷ Per standard convention I denote the size of a set using the corresponding roman symbol. E.g. E is the size of \mathcal{E} .

of a KBC system is to rank the elements of \mathcal{F}^c so that facts that are correct receive a smaller rank than incorrect facts.

Embedding Model: I assume that every relation $r \in \mathcal{R}$ and entity $e \in \mathcal{E}$ can be represented using real valued vectors $\mathbf{r} \in \mathbb{R}^d$ and $\mathbf{e} \in \mathbb{R}^{\tilde{d}}$; d and \tilde{d} may have different values. The vector representation of each tuple t is computed from its constituent entities via a composition function $c : \mathbb{R}^{\tilde{d}} \times \mathbb{R}^{\tilde{d}} \rightarrow \mathbb{R}^d$, i.e. $\mathbf{t} = c(\mathbf{e}, \mathbf{e}')$. For example c may denote vector concatenation, in which case $\mathbf{t} = [\mathbf{e}^T, \mathbf{e}'^T]^T$. I will use the semicolon symbol ; as an infix operator to denote vector concatenation, i.e. $(\mathbf{x}; \mathbf{y}) = [\mathbf{x}^T, \mathbf{y}^T]^T$. Finally, $\mathbf{x} \geq \mathbf{y}$ denotes that the vector \mathbf{x} is elementwise larger than \mathbf{y} and $B(\mathbf{x}, r)$ denotes the L_2 ball centered at \mathbf{x} with radius r .

Score Function: A majority of the existing work on embedding based KBC measures the *correctness* of a fact via a scoring function, $\text{score} : \mathcal{R} \times \mathcal{E} \times \mathcal{E} \rightarrow \mathbb{R}$, with the property that when $\text{score}(f) > \text{score}(f')$, fact f is more likely to be correct than f' . The two major classes of score functions are:

$$\text{score}(f) = \langle \mathbf{r}, \mathbf{t} \rangle \quad (5.2)$$

$$\text{score}(f) = -||\mathbf{r} - \mathbf{t}||^2 \quad (5.3)$$

In Equations (5.2–5.3), \mathbf{r} and \mathbf{t} are vector representations of r and $t = (e, e')$, respectively, that are constituents of $f = (r, (e, e'))$. For brevity, I will omit this expansion from here onwards.

Unconstrained Objectives for Learning Score Function (Rendle et al. 2009) proposed the Bayesian Personalized Ranking (BPR) objective as a way of tuning recommendation systems when a user can only observe positive training data, such as correct facts, but the facts that are absent may be either

correct or incorrect. in this chapter I will focus on the BPR objective since this objective has been used for learning the parameters of a KBC system by various researchers (Rendle et al. 2009; Demeester, Rocktäschel, and Riedel 2016; Riedel et al. 2013). (Wang and Cohen 2016) experimentally showed that the BPR objective outperforms other objectives such as Hinge Loss and Log Loss.

BPR posits that the training data is a single joint sample of $U(U-1)$ bernoulli random variables $\{b_{ff'} \mid f \in \mathcal{U}, f' \in \mathcal{U}, f' \neq f\}$. $b_{ff'}$ equals 1 when f is in \mathcal{F} and f' is in \mathcal{F}^c and 0 otherwise. $b_{ff'}$ is parameterized by its probability $p_{ff'}$ and all $b_{ff'}$ are conditionally independent given the probabilities $p_{ff'}$. The probability values must obey the reasonable condition that $p_{ff'} = 1 - p_{f'f}$. A natural way to satisfy this condition is to parameterize $p_{ff'}$ as $\sigma(\text{score}(f) - \text{score}(f'))$ where σ is the sigmoid function.⁸ The BPR estimator is simply the L_2 regularized MLE estimator of this probabilistic model, with regularization strength α . Table 5.4 lists instances of the BPR objective that arise with different score functions.

Logical Consistency of Embeddings through Constraints My general scheme for incorporating logical relations into embeddings is to ensure that during the learning of the vector representation of entities and relations, the score of a consequent fact will be greater than the score of any of its antecedents. In other words if $f_1, \dots, f_{n-1} \implies f_n$ then $\text{score}(f_n) \geq \min_{i \in [1, n-1]}(\text{score}(f_i))$. If this does not hold, then it will be possible for my KB to assign a low score to a logically entailed fact even though all of its antecedents have a high score.

I analyzed common logical rules found in large scale KBs, and for different combinations of a logical rule and scoring function, I devised inequalities that

⁸The sigmoid function, $\sigma(x) = \frac{1}{1+\exp(-x)}$, has the useful properties that $\sigma(x) + \sigma(-x) = 1$ and $\frac{d\sigma(x)}{dx} = \sigma(x)\sigma(-x)$.

Model	score	Minimization Objective (J)
$\frac{A}{R} \mid t = (\mathbf{e}; \mathbf{e}')$	(5.2)	$-\sum_{(f, \bar{f}) \in \mathcal{F} \times \mathcal{F}^c} \log(\sigma(\langle \mathbf{r}, \mathbf{t} \rangle - \langle \bar{\mathbf{r}}, \bar{\mathbf{t}} \rangle)) + \alpha(\sum_{r \in \mathcal{R}} \ \mathbf{r}\ ^2 + \sum_{t \in \mathcal{E}} \ \mathbf{e}\ ^2)$
$\frac{B}{t = (\mathbf{e}; \mathbf{e}'; \mathbf{e}^T \mathbf{e}')$	(5.2)	$-\sum_{(f, \bar{f}) \in \mathcal{F} \times \mathcal{F}^c} \log \sigma \left(\begin{array}{c} \langle \mathbf{r}_1, \mathbf{e} \rangle + \langle \mathbf{r}_2, \mathbf{e}' \rangle + \langle \mathbf{e}, \mathbf{e}' \rangle \\ - \langle \bar{\mathbf{r}}_1, \bar{\mathbf{e}} \rangle - \langle \bar{\mathbf{r}}_2, \bar{\mathbf{e}}' \rangle - \langle \bar{\mathbf{e}}, \bar{\mathbf{e}}' \rangle \end{array} \right) + \alpha(\sum_{r \in \mathcal{R}} \ \mathbf{r}\ ^2 + \sum_{t \in \mathcal{E}} \ \mathbf{e}\ ^2)$
$\frac{C}{T} \mid t = (\mathbf{e}; \mathbf{e}')$ $\frac{T}{T} \mid t = \mathbf{e} - \mathbf{e}'$	(5.3)	$-\sum_{(f, \bar{f}) \in \mathcal{F} \times \mathcal{F}^c} \log(\sigma(\ \bar{\mathbf{r}} - \bar{\mathbf{t}}\ ^2 - \ \mathbf{r} - \mathbf{t}\ ^2)) + \alpha(\sum_{r \in \mathcal{R}} \ \mathbf{r}\ ^2 + \sum_{t \in \mathcal{E}} \ \mathbf{e}\ ^2)$
$\frac{D}{t = (\mathbf{e}; \mathbf{e}' \parallel \mathbf{e} - \mathbf{e}' \parallel)}$	(5.3)	$-\sum_{(f, \bar{f}) \in \mathcal{F} \times \mathcal{F}^c} \log \sigma \left(\begin{array}{c} \ \bar{\mathbf{r}}_1 - \bar{\mathbf{e}}\ ^2 + \ \bar{\mathbf{r}}_2 - \bar{\mathbf{e}}'\ ^2 + \ \bar{\mathbf{e}} - \bar{\mathbf{e}}'\ ^2 \\ - \ \mathbf{r}_1 - \mathbf{e}\ ^2 - \ \mathbf{r}_2 - \mathbf{e}'\ ^2 - \ \mathbf{e} - \mathbf{e}'\ ^2 \end{array} \right) + \alpha(\sum_{r \in \mathcal{R}} \ \mathbf{r}\ ^2 + \sum_{t \in \mathcal{E}} \ \mathbf{e}\ ^2)$

Table 5.4: Instances of the BPR objective corresponding to different choices of composition and score functions. For example, if $c(\mathbf{e}, \mathbf{e}') = (\mathbf{e}; \mathbf{e}')$ and Eq. 5.2 is used as the score function then I need to minimize the function in the first row with respect to \mathbf{r}, \mathbf{e} . In the first and third row, $\mathbf{r} = (\mathbf{r}_1; \mathbf{r}_2)$, in the second row $\mathbf{r} = (\mathbf{r}_1; \mathbf{r}_2; 1)$ and in last row $\mathbf{r} = (\mathbf{r}_1; \mathbf{r}_2; 0)$. The symbol \otimes refers to the vector outer product operator that takes two vectors of size \tilde{d} and produces a vector of size \tilde{d}^2 . Since this scoring function is equivalent to the score function of RESCAL I call the model R. Similarly the scoring function for T is the same as TransE (Bordes et al. 2013).

the score function should satisfy. I translated those inequalities into constraints that restrict the entity and relation representations in a KB.

I use the projected subgradient descent algorithm for learning the parameters of my KBS system. Algorithm 2 shows a specific instance, for Model A and batch size 1, of my parameter learning algorithm with a general set of rules \mathcal{L} . I now show how to construct convex constraints from logical rules.

5.4.1 Constraints for Logical Consistency: Relational Implication

I now present the constraints for guaranteeing that the predictions from embeddings based KBC systems are consistent with logical rules starting with implication rules. An implication rule of the form, $\text{RELIMP}(r, r')$, specifies that if a fact $f = (r, t)$ is correct, then (r', t) must also be correct. For example, the rule $\text{RELIMP}(\text{HusbandOf}, \text{SpouseOf})$ enforces that if my KB predicts the fact, $(\text{HusbandOf}, (\text{Don}, \text{Mel}))$, then it will also predict $(\text{SpouseOf}, (\text{Don}, \text{Mel}))$. As explained above I can enforce such a rule by ensuring that $\text{score}(r', t) \geq \text{score}(r, t) \forall t \in \mathcal{T}$.

9

When I use the inner product score function (5.2) then this inequality can be enforced by ensuring that $\langle \mathbf{r}' - \mathbf{r}, \mathbf{t} \rangle \geq 0$ for all $t \in \mathcal{T}$. I constrain \mathbf{t} to lie in a subset of \mathbb{R}^d , say \mathbb{T} , then the implication rule can be enforced by constraining $\mathbf{r}' - \mathbf{r}$ to lie in the dual cone of \mathbb{T} , denoted \mathbb{T}^* . A very convenient special case arises when I chose \mathbb{T} to be a “self dual cone” for which $\mathbb{T} = \mathbb{T}^*$. The set of positive real vectors \mathbb{R}_+^d is one example of such a self-dual cone. ¹⁰

⁹I abuse notation in saying that $\text{score}(r, (x, y)) = \text{score}(r, x, y)$.

¹⁰Other self-dual cones, distinct from \mathbb{R}_+^d also exist such as the Lorentz cone $\{x \in \mathbb{R}^d \mid x_d \geq \sqrt{\sum_{i=1}^{d-1} x_i^2}\}$. I refer the reader to (Gruber 2007) for more details on the geometry of closed

When I use the L_2 distance score function (5.3) then the restriction on the score function translates into the following constraints on the vector representations: $\|\mathbf{r} - \mathbf{t}\|^2 - \|\mathbf{r}' - \mathbf{t}\|^2 \geq 0 \implies \langle \mathbf{r} - \mathbf{r}', \mathbf{r}' + \mathbf{r} - 2\mathbf{t} \rangle \geq 0 \implies \langle \mathbf{r} - \mathbf{r}', \mathbf{r}' + \mathbf{r} \rangle / 2 \geq h_{\mathbb{T}}(\mathbf{r} - \mathbf{r}')$. Here $h_{\mathbb{T}}(\mathbf{x})$ is the value of the support function of \mathbb{T} at \mathbf{x} which is defined as $h_{\mathbb{T}}(\mathbf{x}) = \sup_{\mathbf{t} \in \mathbb{T}} \langle \mathbf{x}, \mathbf{t} \rangle$. It is necessary and sufficient for the feasibility of this constraint that the $h_{\mathbb{T}}$ function should be finite for at least one value of $\mathbf{x} = \mathbf{r} - \mathbf{r}'$. Once I have fixed $\mathbf{r} - \mathbf{r}'$ then $\mathbf{r} + \mathbf{r}'$ can be easily chosen from the halfspace $H^-(\mathbf{r}' - \mathbf{r}, 2h_{\mathbb{T}}(\mathbf{r} - \mathbf{r}'))$. Note that if $h_{\mathbb{T}}$ is difficult to compute then implementing this constraint will also be difficult, therefore I must chose \mathbb{T} wisely.

One example of a good choice of \mathbb{T} is \mathbb{R}_+^d . $h_{\mathbb{R}_+^d}(\mathbf{r} - \mathbf{r}')$ is finite and zero iff $\mathbf{r} - \mathbf{r}' \leq \mathbf{0}$ therefore, the value of $\mathbf{r} + \mathbf{r}'$ must lie in the halfspace $\langle \mathbf{r} - \mathbf{r}', \mathbf{r}' + \mathbf{r} \rangle \geq 0$. Unfortunately, the problem of finding \mathbf{r} and \mathbf{r}' vectors that satisfy this constraint is non-convex and it is not possible to project on to this set given a pair of vectors that violate the constraints. I remedy this situation by adding an additional constraint that $\mathbf{r} + \mathbf{r}'$ must also lie in the negative orthant, i.e. $\mathbf{r} + \mathbf{r}' \leq \mathbf{0}$. Table 5.5 presents all the derived constraints. Unfortunately, since the \mathbb{T} model defines $\mathbf{t} = \mathbf{e} - \mathbf{e}'$, therefore it is not possible to set $\mathbb{T} = \mathbb{R}_+^d$. In the case of the \mathbb{T} model if I constrain \mathbf{e} to lie in $B(\mathbf{0}, \rho)$ then \mathbf{t} must lie in $B(\mathbf{0}, 2\rho)$ and $h_{\mathbb{T}}(\mathbf{r} - \mathbf{r}') = 2\rho(\mathbf{r} - \mathbf{r}') \implies \frac{\langle \mathbf{r} - \mathbf{r}', \mathbf{r}' + \mathbf{r} \rangle}{\|\mathbf{r} - \mathbf{r}'\|} \geq 4\rho$. One way to make this constraint amenable to efficient projection is to enforce that $\mathbf{r} + \mathbf{r}' = 4\rho(\mathbf{r} - \mathbf{r}')$ and $\|\mathbf{r} - \mathbf{r}'\|^2 \geq 1 \implies \|\mathbf{r}'\|^2 \geq |2\rho - .5|$. This constraint becomes trivial if $\rho = 0.25$

convex cones and their polar and dual sets.

5.4.1.1 Reverse Relational Implication and Symmetry

A reverse relational implication rule denoted by $\text{REVIMP}(r, r')$ specifies that if $(r, (x, y))$ is correct, then $(r', (y, x))$ is also correct for all $(x, y) \in \mathcal{T}$. This rule can be enforced through the inequality that $\text{score}(r', y, x) \geq \text{score}(r, x, y)$. Depending on the model let $\mathbf{r} = (\mathbf{r}_1; \mathbf{r}_2)$ or $(\mathbf{r}_1; \mathbf{r}_2; 1 \text{ or } 0)$ as shown in Table 5.4, and similarly decompose \mathbf{r}' . I will omit this detail in later sections. Under models A and B, this inequality translates to the following constraint $\langle \mathbf{y}, \mathbf{r}'_1 \rangle + \langle \mathbf{x}, \mathbf{r}'_2 \rangle \geq \langle \mathbf{x}, \mathbf{r}_1 \rangle + \langle \mathbf{y}, \mathbf{r}_2 \rangle$ and under models C and D, the necessary constraints are $\langle \mathbf{r}_1 - \mathbf{r}'_2, \mathbf{r}_1 + \mathbf{r}'_2 - 2\mathbf{x} \rangle \geq \langle \mathbf{r}'_1 - \mathbf{r}_2, \mathbf{r}'_1 + \mathbf{r}_2 - 2\mathbf{y} \rangle$. Stronger versions of these constraints, which are more efficient to enforce, are shown in Table 5.5.

A symmetry rule denoted as $\text{SYMM}(r)$ specifies that if the fact $(r, (e, e'))$ is known to be correct then $(r, (e', e))$ is also correct. I can only comply with this logical rule in an embedding base KB by ensuring that $\text{score}(r, e, e') = \text{score}(r, e', e)$. Under all 4 score models this rule can be enforced only by ensuring that $\mathbf{r}_1 = \mathbf{r}_2$.

5.4.1.2 Entailment

A type A entailment logical rule denoted as $\text{ENTAIL}_A(r, e, r', e')$ specifies that $(r, (e, x))$ implies $(r', (e', x))$ for all x in \mathcal{E} .¹¹ A Type B entailment rule, $\text{ENTAIL}_B(r, e, r', e')$ specifies that $(r, (x, e))$ implies $(r', (x, e'))$. r and r' may denote the same relation. For example, the rule $\text{ENTAIL}_B(\text{IsA}, \text{Man}, \text{IsA}, \text{Mortal})$ can be used to enforce that if $(\text{IsA}, (\text{Socrates}, \text{Man}))$ then the KB must also predict that $(\text{IsA}, (\text{Socrates}, \text{Mortal}))$. The final constraints required to implement

¹¹I use the term, entailment, in the sense of entailment of properties. Note that this is different from implication.

a type B entailment rule are shown in Table 5.6.¹²

5.4.1.3 Property Transitivity

A property transitivity rule denoted $\text{PROTRANS}(r, r', e', r'', e'')$ specifies that if $(r, (x, y))$ and $(r', (y, e'))$ are true then $(r'', (x, e''))$ is also true. For example, the rule $\text{PROTRANS}(\text{Partner}, \text{Convicted}, \text{Criminal}, \text{Suspected}, \text{Criminal})$ can be used to incorporate the common sense rule that if an entity is the partner of a convicted criminal then it will be suspected of being a criminal into the embeddings based KB. Note that the score of the hypothesis fact $(r'', (x, e''))$ should be high if the antecedent facts have high score for any possible entity y . A natural way in which I can incorporate such a rule into score based KBC models is by ensuring that $\text{score}(r'', x, e'') \geq \max_{y \in \mathcal{E}} \min(\text{score}(r, x, y), \text{score}(r', y, e'))$. In order to derive efficient constraints that can enforce this inequality I now strengthen the constraint imposed on the score function by replacing the min function in the lower bound to a convex combination of the scores, i.e. let $\lambda \in (0, 1)$, I enforce the inequality that $\text{score}(r'', x, e'') \geq \max_{y \in \mathcal{E}} \lambda \text{score}(r, x, y) + (1 - \lambda) \text{score}(r', y, e')$.

Since a convex combination of two values is greater than their minimum, this stronger inequality translates to the following constraint for model A:

$$\langle \mathbf{e}'', \mathbf{r}_2'' \rangle - (1 - \lambda) \langle \mathbf{e}', \mathbf{r}_2' \rangle + \langle \mathbf{x}, \mathbf{r}_1'' - \lambda \mathbf{r}_1 \rangle \geq \langle \mathbf{y}, \lambda \mathbf{r}_2 + (1 - \lambda) \mathbf{r}_1' \rangle. \text{ Let } \mathbf{a} = \frac{\mathbf{r}_1'' - \lambda \mathbf{r}_1 + \mathbf{e}''}{\lambda},$$

$$\mathbf{b} = -\frac{(1 - \lambda)(\mathbf{r}_1' + \mathbf{e}') + \lambda \mathbf{r}_1}{\lambda}, \mathbf{c} = \frac{\langle \mathbf{r}_2'', \mathbf{e}'' \rangle - (1 - \lambda) \langle \mathbf{r}_2', \mathbf{e}' \rangle}{\lambda}, \text{ and let } \mathbb{E} \text{ contain the set } \{\mathbf{e} \mid e \in \mathcal{E}\}.$$

For Model B, the above inequality on the score function leads to the the constraint:

¹²Details: To implement a type B entailment rule I need to ensure that $\text{score}(r', x, e') \geq \text{score}(r, x, e)$ for all $x \in \mathcal{E}$. Under model A this inequality translates to, $\langle \mathbf{r}_1' - \mathbf{r}_1, \mathbf{x} \rangle \geq \langle \mathbf{r}_2, \mathbf{e} \rangle - \langle \mathbf{r}_2', \mathbf{e}' \rangle$. Model B requires $\langle \mathbf{r}_1' - \mathbf{r}_1 + \mathbf{e}' - \mathbf{e}, \mathbf{x} \rangle \geq \langle \mathbf{r}_2, \mathbf{e} \rangle - \langle \mathbf{r}_2', \mathbf{e}' \rangle$. Model C requires $\langle \mathbf{r}_1 - \mathbf{r}_1', \mathbf{r}_1 + \mathbf{r}_1' - 2\mathbf{x} \rangle \geq \langle \mathbf{r}_2' - \mathbf{e}' + \mathbf{r}_2 - \mathbf{e}, \mathbf{r}_2' - \mathbf{e}' - \mathbf{r}_2 + \mathbf{e} \rangle$, and finally the constraints over model D's score functions are $\langle \mathbf{r}_1 - \mathbf{r}_1', \mathbf{r}_1 + \mathbf{r}_1' \rangle + \langle \mathbf{e} - \mathbf{e}', \mathbf{e} + \mathbf{e}' \rangle - \langle \mathbf{r}_2' - \mathbf{e}' - \mathbf{r}_2 + \mathbf{e}, \mathbf{r}_2' - \mathbf{e}' + \mathbf{r}_2 - \mathbf{e} \rangle \geq 2\langle \mathbf{r}_1 - \mathbf{r}_1' - \mathbf{e} - \mathbf{e}', \mathbf{x} \rangle$.

$\forall \mathbf{x}, \mathbf{y} \in \mathbb{E}, \langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{x}, \mathbf{a} \rangle + \langle \mathbf{y}, \mathbf{b} \rangle + c$. Remember that my goal is to devise a set \mathbb{E} , and constraints on relation embeddings such that it is efficient to project onto it and for which the above inequality can be guaranteed. The following proposition shows how to construct such a set:

Proposition 5. *Let \mathbf{x}, \mathbf{y} be members of $\mathbb{R}_+^d \cap B(\mathbf{a}, \|\mathbf{a}\|)$ and $\mathbf{a} \geq \mathbf{0}$ then $\langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{x} + \mathbf{y}, \mathbf{a} \rangle$.*

The above proposition shows that if $\mathbf{a} = \mathbf{b}$ and $c \geq 0$ then by setting $\mathbb{E} = \mathbb{R}_+^d \cap B(\mathbf{a}, \|\mathbf{a}\|)$ I can satisfy the above constraints.

Rule	Model	Constraints
RELIMP(r, r')	A, R, B	$\mathbf{r} \leq \mathbf{r}'$
	C, D	$\mathbf{r} \leq \mathbf{r}' \leq -\mathbf{r}$
REVIMP(r, r')	A, B	$\mathbf{r}'_2 \geq \mathbf{r}_1, \mathbf{r}'_1 \geq \mathbf{r}_2$
	C, D	$\mathbf{r}_1 \leq \mathbf{r}'_2 \leq -\mathbf{r}_1, \mathbf{r}_2 \leq \mathbf{r}'_1 \leq -\mathbf{r}_2$.
	R	$matrix(\mathbf{r}') \geq matrix(\mathbf{r})$

Table 5.5: Constraints sufficient for enforcing RELIMP(r, r') and REVIMP(r, r') The constraint $\mathbf{e} \geq \mathbf{0} \forall e \in \mathcal{E}$ applies for all models. *matrix* is the inverse of the operation that converts a matrix to a vector by concatenating its columns. I.e. *matrix*(\mathbf{r}) denotes the matrix form of the vector \mathbf{r} .

Alg. 2 Projected SGD for Model A, Batch Size=1

Given: $\mathcal{F}, \mathcal{F}^c, \mathcal{L}$. Hyperparameters: α, η, S .

```
for each fact  $f \in \mathcal{F}$  do
  for S steps do
    Sample  $\bar{f} = (\bar{t}, \bar{r})$  from  $\mathcal{F}^c$ 
    Let  $v = \sigma(\langle \bar{\mathbf{r}}, \bar{\mathbf{t}} \rangle - \langle \mathbf{r}, \mathbf{t} \rangle)$ 
    ▷ Fix  $\mathbf{e}$  and optimize  $J$ 
     $\frac{\partial J^{(f)}}{\partial \mathbf{r}} = -\mathbf{t}v, \frac{\partial J^{(f)}}{\partial \bar{\mathbf{r}}} = \bar{\mathbf{t}}v$ 
     $(\mathbf{r}; \bar{\mathbf{r}}) \leftarrow \text{proj}_{\mathcal{L}} \left( (\mathbf{r}; \bar{\mathbf{r}}) - \eta \left( \left( \frac{\partial J^{(f)}}{\partial \mathbf{r}}; \frac{\partial J^{(f)}}{\partial \bar{\mathbf{r}}} \right) + 2\alpha(\mathbf{r}; \bar{\mathbf{r}}) \right) \right)$ 
    ▷ Fix  $\mathbf{r}$  and optimize  $J$ 
     $\frac{\partial J^{(f)}}{\partial \mathbf{t}} = -\mathbf{r}v, \frac{\partial J^{(f)}}{\partial \bar{\mathbf{t}}} = \bar{\mathbf{r}}_1 v$ 
     $(\mathbf{t}; \bar{\mathbf{t}}) \leftarrow \text{Proj}_{\mathcal{L}} \left( (\mathbf{t}; \bar{\mathbf{t}}) - \eta \left( \left( \frac{\partial J^{(f)}}{\partial \mathbf{t}}; \frac{\partial J^{(f)}}{\partial \bar{\mathbf{t}}} \right) + 2\alpha(\mathbf{t}; \bar{\mathbf{t}}) \right) \right)$ 
  end for
end for
```

5.4.1.4 Type Implication

A type implication rule, denoted as $\text{TYPEIMP}(r, e, r')$, specifies that if the fact $(r, (x, y))$ is correct then $(r', (x, e))$ is also correct $\forall (x, y) \in \mathcal{T}$. In other words, this rule enforces that positional arguments of a relation possess certain properties. For example, the rule $\text{TYPEIMP}(\text{Husband of}, \text{Man}, \text{Gender})$ can enforce that if my KB predicts the fact that $(\text{Husband of}, (\text{Don}, \text{Mel}))$ then it also predicts that $(\text{Gender}, (\text{Don}, \text{Man}))$.

Under model A the $\text{TYPEIMP}(r, e, r')$ rule translates to the following inequality for the parameters $\langle \mathbf{x}, \mathbf{r}'_1 \rangle - \langle \mathbf{x}, \mathbf{r}_1 \rangle \geq \langle \mathbf{y}, \mathbf{r}_2 \rangle - \langle \mathbf{e}, \mathbf{r}'_2 \rangle \forall (x, y) \in \mathcal{T}$. Let $\mathbf{a} = \mathbf{e} + \mathbf{r}'_1 - \mathbf{r}_1$, $\mathbf{b} = -\mathbf{r}_2$ and $c = \langle \mathbf{r}'_2, \mathbf{e} \rangle$. Under model B, the restriction on the score function translates to: $\langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{a}, \mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{x} \rangle + c$. The analysis for this case again relies on Proposition 5 and the analysis for models C and D is yet out of reach. See Table 5.6.

Model	Constraints
A	$\mathbf{r}'_1 \geq \mathbf{r}_1, \langle \mathbf{r}_2, \mathbf{e} \rangle \leq \langle \mathbf{r}'_2, \mathbf{e}' \rangle$
B	$\mathbf{r}'_1 \geq \mathbf{r}_1 + \mathbf{e} - \mathbf{e}', \langle \mathbf{r}_2, \mathbf{e} \rangle \leq \langle \mathbf{r}'_2, \mathbf{e}' \rangle$
C	$\mathbf{r}_1 \leq \mathbf{r}'_1 \leq -\mathbf{r}_1, \mathbf{r}_2 - \mathbf{e} \leq \mathbf{r}'_2 - \mathbf{e}',$ $\mathbf{r}'_2 + \mathbf{r}_2 \leq \mathbf{e}' + \mathbf{e}$
D	$\mathbf{r}_1 - \mathbf{r}'_1 \leq \mathbf{e}' - \mathbf{e}, \mathbf{e} \geq \mathbf{e}', \mathbf{r}_1 \leq \mathbf{r}'_1 \leq -\mathbf{r}_1,$ $\mathbf{r}_2 - \mathbf{e} \leq \mathbf{r}'_2 - \mathbf{e}', \mathbf{r}'_2 + \mathbf{r}_2 \leq \mathbf{e}' + \mathbf{e}$

Table 5.6: Sufficient constraints for $\text{ENTAIL}_B(r, e, r', e')$. The constraint $\mathbf{e} \geq \mathbf{0} \forall e \in \mathcal{E}$ applies for all models.

Rule	Model	Constraints
PROTRANS	A	$\mathbf{r}''_1 \geq \lambda \mathbf{r}_1, \lambda \mathbf{r}_2 + (1 - \lambda) \mathbf{r}'_1 \leq \mathbf{0}$ $\langle \mathbf{e}'', \mathbf{r}''_2 \rangle \geq (1 - \lambda) \langle \mathbf{e}', \mathbf{r}'_2 \rangle$
	B	$\mathbf{r}''_1 + \mathbf{e}'' + (1 - \lambda)(\mathbf{r}'_1 + \mathbf{e}') = \mathbf{0},$ $\langle \mathbf{r}''_2, \mathbf{e}'' \rangle \geq (1 - \lambda) \langle \mathbf{r}'_2, \mathbf{e}' \rangle$, and $\mathbf{a} \geq \mathbf{0}, \forall x \in \mathcal{E}, \mathbf{x} \in \mathbb{R}_+^d \cap B(\mathbf{a}, \ \mathbf{a}\)$
TYPEIMP	A	$\mathbf{r}'_1 \geq \mathbf{r}_1, \langle \mathbf{e}, \mathbf{r}'_2 \rangle \geq 0$ and $\mathbf{r}_2 \leq \mathbf{0}$
	B	$\mathbf{e} + \mathbf{r}'_1 = \mathbf{r}_1 - \mathbf{r}_2, \langle \mathbf{r}'_2, \mathbf{e} \rangle > 0$, and $-\mathbf{r}_2 \geq \mathbf{0}, \forall x \in \mathcal{E} \mathbf{x} \in \mathbb{R}_+^d \cap B(-\mathbf{r}_2, \ \mathbf{r}_2\)$

Table 5.7: Constraints for enforcing $\text{PROTRANS}(r, r', e', r'', e'')$ and $\text{TYPEIMP}(r, e, r')$.
 $\mathbf{a} = \frac{\mathbf{r}''_1 - \lambda \mathbf{r}_1 + \mathbf{e}''}{\lambda}$

5.4.2 Evaluating Logical Deduction and KBC on Word-Net

My method for training embeddings based KBC systems allows for a very interesting application for solving logical puzzles using an embedding based KBC system without using an external logical-symbolic subsystem. I perform a controlled experiment where I compare the performance of an embedding based KBC system trained with the constraints versus a system that has been trained without those constraints.

Data Consider the logical deduction problem shown in Table 5.8. This is a simplified version of a logical puzzle presented in (Russell, Norvig, and Intelligence 1995). In this puzzle, Nono is a country that possesses a *WMD* and **Benedict** has traded with Nono. The KB has to deduce whether **Benedict** is a criminal based on just two input facts and 3 rules. The total number of facts is $5^2 \times 4 = 100$.

Rules
RELIMP(TradeWith, TransactWith)
ENTAIL _B (Possess, WMD, Considered, Enemy)
PROTRANS(TransactWith, Enemy, Considered, Criminal, Considered)
Facts
(Possess, (Nono, WMD))
(TradeWith, (Benedict, Nono))
Query ?
(Considered, (Benedict, Criminal))

Table 5.8: A Logical Deduction Problem. Based on the rules and facts a KB should infer that **Benedict** is a **Criminal**.

Evaluation I train two versions of two KBC systems, Models A and B, with batch size= 1, $\alpha = 0.001$, $\eta = 0.1$, $S = 200$, $d = 50$, and $\tilde{d} = 25$ using

Model	Baseline			ELKB		
	P@10	MRR	MAP	P@10	MRR	MAP
A	0.00	0.02	0.01	0.20[†]	0.44[†]	0.83[†]
B	0.00	0.03	0.03	0.17[†]	0.26[†]	0.35[†]

Table 5.9: Table of Results. The baseline of R is equivalent to the RESCAL method. Bold marks that the average performance is higher. [†] implies that the difference is significant with two-tailed p-value ≤ 0.005 as measured by a matched pair t-test.

Algorithm 2. Both KBs were trained in one pass using the two training facts. The only difference was that the baseline system did not constrain the embeddings to obey logically derived geometric constraints. After training, I queried the KBs for the scores of all possible facts. I ranked all the facts based on their scores, excluding the training facts, and marked all facts that could be logically entailed from the two training facts as correct results and the rest of them as incorrect. I performed 10 runs, and in each run, I computed the MRR, P@10, MAP for the two models. Finally, I averaged these quantities over 10 runs.

Results Table 5.9 shows that my method was able to rank logically entailed facts with much higher precision and recall than the baseline systems. This validates my intuition that logical rules can be usefully incorporated into the parameter learning mechanism of a KBC system via simple geometric constraints even for low dimensional embeddings. The reason for the large improvement in performance by the ELKB system in comparison to the baseline is that the ELKB model makes the score of entailed facts higher than the score of non-entailed facts because of the constraints during learning. E.g. the scores of entailed facts such as `(Considered, (Nono, Enemy))`, and `(TransactWith, (Benedict, Nono))` are forced to be high in comparison to non-entailed facts such as `(Trade With, (Benedict, WMD))`. In comparison, the baseline method does not have this

systematic advantage, and its scores remain unchanged.

5.4.3 Evaluating Link Prediction on WordNet

In the link prediction task, the KBC system is given incomplete facts, with either a missing head entity or tail entity, i.e. given either $(r, (_, e'))$, or $(r, (e, _))$ the system has to predict e or e' respectively. I evaluated the utility of proposed constraints by comparing the performance of model A and model B trained with and without the constraints. I now present the results of my experiments on the WN18 knowledge graph,¹³ derived from WordNet, and released by (Bordes et al. 2013), which is a popular testbed for KBC algorithms (Wang et al. 2014; Lin et al. 2015; Toutanova et al. 2015; Yang et al. 2015).

Data The WN18 dataset comes with standard train, development and test splits. These three splits of the data contain 141442, 5000, and 5000 facts respectively. The total number of relations in the WN18 dataset is 18, and the number of entities is 40,943. Recently (Guo et al. 2016) publicly released a list of logical rules¹⁴

For all the models I fixed batch size= 10, $\alpha = 0.001$, $\eta = 0.125$, $S = 200$, $\tilde{d} = 100$, for model T , $d = 100$ and otherwise $d = 200$. Following existing work, I measured the MRR, HITS@3 and Hits@10 metrics and reported their average over the two tasks of head entity prediction and tail entity prediction. Instead of training in a single pass, I trained my models for 50 epochs on the WN18 dataset and chose the best parameters using early stopping on the validation set. In other words, I used the parameters from that epoch which performed the

¹³I found that the performance of models C , D and R was too low therefore I do not report their results.

¹⁴aclweb.org/anthology/attachments/D/D16/D16-1019.Attachment.zip

Model	Project	MRR	HITS@3	HITS@10
A	No	0.0152	0.016	0.0330
A	Yes	0.0238	0.03	0.0514
B	No	0.0677	0.072	0.137
B	Yes	0.241	0.283	0.50
T	No	0.311	0.412	0.66
$B^{\text{project}}+T$	–	0.367	0.475	0.708

Table 5.10: MRR, HITS@3 and HITS@10 of the constrained and unconstrained versions of models A, B and unconstrained T. B+T reports the results of combining models B and model T.

best on the validation set in terms of the HITS@10 metric. Finally, I combined the predictions of the best performing T model and the best performing system based on model B. In order to combine the two ranking systems I trained a logistic regression classifier using the default settings in `vowpal wabbit`¹⁵ to first predict whether model T or model B will produce a better ranking and then output that system’s ranking over entities for evaluation. My logistic regression classifier had 73% accuracy on the training data and 70% accuracy over the test data. By using this third system, I were able to create a single ranking system that performed better than model T which is very similar to the TransE model.¹⁶

Results Table 5.10 shows that both the constrained and unconstrained versions of model A perform quite poorly. This is to be expected since model A scores a triplet $(r, (e, e'))$ as $\langle \mathbf{r}, \mathbf{e} \rangle + \langle \mathbf{r}, \mathbf{e}' \rangle$. Regardless of e' , the ranking produced by the model will remain the same. Therefore model A is unsuitable for this

¹⁵https://github.com/JohnLangford/vowpal_wabbit

¹⁶The main differences between model T and TransE are that TransE used hinge loss versus the BPR objective. TransE does not regularize the relation embeddings and forces the entity embeddings to lie on the unit sphere, instead in model T I add a quadratic regularization term to regularize the embeddings.

task, for similar reason model C is also an unsuitable model. However, the drastic improvement in the performance of model B when it is trained according to the constraints corresponding to the REVIMP rules demonstrates the utility of my proposed constraints. After adding the constraints, the MRR increased almost 3 times and the value of HITS@10 by 4 times from 0.137. Recall that at test/inference time the constraints do not play any role, so the only role of the constraints is as a form of regularization on the parameters of the model.

5.4.4 Discussion

It is instructive to look at a few examples of the predictions that model B makes and to compare them to the predictions made by model T. Table 5.11 compares the top 5 predictions of constrained model B with the predictions from model T for the input (`hypernym, (_, floweringshrubNN_1)`). The true answer is `poinciana_gilliesii_NN_1` so the model T achieves a reciprocal rank of 1 for this example, but constrained model B is not able to rank the right answer within the top 5 answers. This list of answers shows that model B ranks those answers higher that are similar to `floweringshrubNN_1`, but it is not able to properly use the relation information. However, by properly combining the models, I can improve the performance of the overall system.

B	T
flowering_shrub_NN_1	poinciana_gilliesii_NN_1
genus_caesalpinia_NN_1	mysore_thorn_NN_1
shrub_NN_1	flowering_shrub_NN_1
tree_NN_1	pernambuco_wood_NN_1
rosid_dicot_genus_NN_1	caesalpinia_bonducella_NN_1

Table 5.11: A comparison of the top 5 predictions of constrained model B with the predictions from model T for the input (`hypernym, (_, flowering shrub NN_1)`).

5.5 Conclusion

I have presented a novel method for incorporating logical constraints into an embedding based knowledge base by constraining the parameters of a KB. My experiments on a small logical deduction problem, and on WordNet, indicate that my ideas of imposing geometric constraints on embeddings for enforcing logical rules are sound and that they can improve the generalization of models that are hard to train otherwise. Although the KBC models A, B, C and D do not perform as well as existing models trained without constraints such as TransE, I show that they can be used as part of a combination of systems to improve upon existing methods.

Chapter 6

Comparative Experiments

In this chapter, I present experiments on the downstream tasks of Coreferent Mention Retrieval and Entity Linking using the MVLSA and Variational Autoencoder representation learning methods.

The Coreferent Mention Retrieval (CMR) task, is an information retrieval task, in which the system receives a query sentence mentioning an entity, and the goal is to retrieve sentences containing coreferent mentions of that entity. A user may use a CMR system to find more mentions of an entity when performing an exploratory task over a corpus containing information about entities. The CMR task is a special case of the well-studied problem of Cross-Document Coreference Resolution (Bagga and Baldwin 1998; Mayfield et al. 2009) – in which the system has to cluster all mentions of all entities – in which, unlike Cross-Doc Coref. Rather than operating on the entire mention graph, the system uses retrieval techniques to limit its focus to an implicit subgraph anchored by the given query mention.

Recently, (Sankepally et al. 2018) introduced the CMR task and introduced a new dataset for this task. In this chapter, I compare a number of unsupervised

representation learning methods to learn representations of mentions. I compare LSA, MVLSA, and Variational Auto Encoder based approaches and demonstrate that these features can improve the performance of a strong information retrieval system.

The Entity Linking task is the task of automatically annotating spans of words in natural language texts that mention an entity with the coreferent entity. Entity linking is also sometimes called entity disambiguation to distinguish it from the task of jointly detecting the span of words mentioning an entity and the linking the span to an entity. In this chapter, I focus exclusively on the task of linking a given span in an unstructured plain text document to an entity in a Knowledge Base.

The ACE corpus (*The ACE 2005 (ACE05) Evaluation Plan* 2005) contains a wide range of documents from varying genres such as newswire and online newsgroups. The entire corpus is annotated with the mention boundaries, coreference information between mentions, the semantic type of each mention, and finally, the entity links for each mention which were added by (Bentivogli et al. 2010). The semantic type of an entity can be from one of seven classes: **Person**, **Organization**, **GPE**, **Location**, **Facility**, **Weapon**, and **Vehicle**. The entity links from mentions to a canonical Wikipedia URL are absent for the **Weapon** and **Vehicle** classes. I compare LSA, MVLSA and Variational Auto Encoder based approaches for entity linking and I observe how these unsupervised features learned can improve entity linking performance.

6.0.1 Hypothesis

Based on the experiments and discussion in chapters 4 and 3 we hypothesize that the NVSE method which is based on the Variational autoencoder framework will outperform the spectral method based MVLSA method for the CMR task. One of the reasons for this is that the MVLSA method requires access to a large number of disparate views which are not readily available for the CMR task.

Due to similar reason, we believe that the VAE based method will be better than MVLSA for the task of entity linking. In fact, we skip testing the MVLSA method in a head-to-head comparison with NVSE on the entity linking task and focus on comparing the VAE based method to a state-of-the-art entity embedding method based on max-margin learning.

6.1 Coreferent Mention Retrieval

The CMR dataset released by (Sankepally et al. 2018) was constructed using the TAC-KBP2014 Entity Discovery and Linking dataset (Ji, Nothman, and Hachey 2014) which is available from the LDC as LDC2014E54, LDC2014E13. I will refer to this collection as TAC14. This dataset contains newswire documents, annotated mentions of entities in those documents, and some entity links between those mentions and canonical URLs of entities in Wikipedia. (Sankepally et al. 2018) used 84 mentions as input queries from TAC14. A subset of the documents from TAC14 collections was chosen for retrieval as follows: First, the earliest and latest dates for the documents from which the query mentions were selected were determined. Then, those documents whose dates did not fall between these dates were filtered out. This reduced the size of the retrieval set from 1

million to 117,132 documents. Given a query mention – out of the 84 mentions – the goal was to find all co-referent mentions for that query out of 117,132 documents. Since the TAC14 collection has fairly sparse mention, annotations, therefore, Sankepally et al. used the Amazon Mechanical Turk platform to obtain additional relevance judgments for a set of candidate mentions. A total of 4,172 relevant mentions were collected in this way.

(Sankepally et al. 2018) measured the performance of a system – at the level of individual sentences – using the Mean Inferred Average Precision metric (Yilmaz and Aslam 2006). The average of infAP over all queries is Mean infAP, and analogously the average of inferred AP is mean Inferred AP. Inferred Average Precision is a refined version of the Average Precision metric that accounts for the fact that some of the results that are returned by a system may actually be relevant but may have been skipped by judges during manual annotation. Inferred Average Precision metric also assumes that the top-K results returned by a system have perfect recall. I briefly explain how inferred Average Precision (infAP) is computed for a query. Let us first recall how Average Precision (AP) is computed. Assume that a retrieval system returns a ranked list of documents, for each document in the ranked list I evaluate whether the document is relevant or not. This gives us a sequence of binary values $(e_i)_{i=1}^N$ where e_i is the binary relevance of the i^{th} result. The average precision is computed as

$$\sum_{i=1}^N \frac{e_i}{N} \frac{\sum_{j=1}^i e_j}{i}$$

Note that $\sum_{j=1}^i e_j / i$ is the precision at the i^{th} position. (Yilmaz and Aslam 2006) showed that the above metric could be considered as an expectation of

the following random experiment.

Algorithm 3 The Average Precision Random Experiment.

- 1: **Input:** A list, L , of binary relevance values, And a map, M , of relevant documents to their rank in L or if a relevant document is not present in L then the default value is L .
 - 2: Select a relevant document at random from the keys of M and let its associated rank be R .
 - 3: Select a rank r uniformly at random from the set $\{1, \dots, R\}$.
 - 4: Output the binary relevance of the document at rank r .
-

Steps 3 and 4 effectively compute the precision at a relevant document, and Step 2 has the effect of weighted averaging over all documents, weighted by the relevance of a document. The inferred Average Precision metric changes steps 3 and 4 to compute precision at rank R more robustly when all the relevant mentions are not available in the retrieved set.

6.1.1 Experiments

I performed mention retrieval experiments on the CMR dataset in a query-by-example setting. I first learned representations of un-linked mention spans of named entities using methods such as MVLSA and Variational Autoencoders, and then I used these features to re-rank the top-100 results returned by the LUCENE (McCandless, Hatcher, and Gospodnetic 2010) Information Retrieval software. In the following sections, I explain the pre-processing steps performed before doing Lucene retrieval and the features used to learn mention representations, and finally how the mention representations were used for the final mention retrieval.

6.1.1.1 Mention Featurization

A mention is a span of words inside a sentence that refers to an entity.¹ Such spans are already marked in the TAC14 collection (Ji, Nothman, and Hachey 2014). Given all the mentions in the corpus, my first step is to obtain the raw features for each mention-span in the corpus. I followed the best-performing pre-processing method from (Sankepally et al. 2018) for the following steps.

I created two sets of features for each mention.² The first field of features – called the *mention field* – uses the mention words, and the second field – called the *document field* – is built upon the background document text for representing candidate mentions. Each field contains binary and real-valued features. The binary features are constructed by counting the occurrence of the mention string, the mention type, trigrams of mention string, and an acronym of the mention. The acronym feature was created by concatenating the first alphabetic character of each mention word. All English stop words such as “the”, “an”, “of” were removed before constructing the features.³ The real-valued features were constructed from the mention words, words from the surrounding sentence and top-scoring words from surrounding document and words in the coreference chain of the mention.⁴ All real-valued features in both the fields were weighted using the BM25 weights⁵ but the binary features were not weighted using BM25.

¹Such Spans are also called named-spans, but I will use the term *mentions* throughout.

²These feature sets are also called fields in the Information Retrieval literature.

³The LUCENE StandardAnalyzer filters the lowercased and normalized output of a grammar-based tokenizer which implements the word-break rules from the Unicode Text Segmentation algorithm (Davis 2011) using a list of English stop words. The normalizer removes the ‘s at the end of words and dots in acronyms.

⁴The coreference chain was extracted using the Stanford CoreNLP coreference resolution system.

⁵See § 4.4.1 for details about BM25.

After obtaining the features, I index them using the LUCENE library. At query time I first retrieve the top-100 mentions from Lucene along with their scores as computed by Lucene. For representing the query, I only used the mention words in the query and projected these query features using LUCENE’s multi-field Query Parser. I.e., I duplicated the mention words across both the mention field and the document field. The *document field* was given a weight of w and the *mention field* was given a weight of 1.0.

For example, the query *Keith Wiggans* is represented as weighted bag-of-words features spread across two fields with associated weights as shown below:

`mention_keith:1.0,document_keith:0.1,mention_wiggans:1.0,document_wiggans:0.1`

6.1.1.2 Learning Entity Embeddings

As mentioned earlier I experimented with MVLSA and the Variational Autoencoder method described in Chapter 4. I conducted experiments on a CMR dataset by (Sankepally et al. 2018) and evaluate the inferred Average Precision metric. The results of the evaluation are shown in Table 6.1 and we can conclude the following from the results:

6.1.2 Results and Discussion

MVLSA m is the intermediate dimensionality, k is the final dimensionality.

6.1.2.1 Performance Comparison between MVLSA and VAE

The highest performance of the Variational Autoencoder is 50.91 infAP points individually which is far superior to the best performance of achieved by MVLSA of 37.08. Clearly the non-linearity of the encoder and optimizing the ELBO

Method	Hyper-parameters	InfAP	Ensemble	
(Sankepally et al. 2018)	$w = 0.01$	54.26		
	$w = 0.01$	54.53		
LSA of Concatenated Views	dim= 20	34.08	50.11	
	dim= 300	40.08	50.32	
	dim= 500	38.08	49.39	
Multiview LSA	$m = 500, k = 500$	37.08	-	
	$m = 500, k = 1000$	36.65	-	
	$m = 500, k = 300$	36.51	-	
	$m = 256, k = 300$	35.08	48.95	
Variational Autoencoder	Multilabel Decoder	$lbv = 0, encdim1 = 150, \beta = 1$	36.35	-
		$lbv = 1, encdim1 = 150, \beta = 1$	39.55	-
		$lbv = 1, encdim1 = 300, \beta = 1$	40.99	-
	Multinomial Decoder	$n_{\text{epoch}} = 5, \beta = 1,$	50.25	55.13
		$n_{\text{epoch}} = 5, \beta = 1, +lbv = 0$	46.29	-
		$n_{\text{epoch}} = 5, \beta = 1, +deduplicate$	50.25	-
		$n_{\text{epoch}} = 0, \beta = 1$	50.74	54.05
		$n_{\text{epoch}} = 40, \beta = 1$	50.29	55.04
		$n_{\text{epoch}} = 5, \beta = 1, +smooth=0.5$	50.91	55.27
		$n_{\text{epoch}} = 2, \beta = 0.0, smooth = 0.5$	49.67	55.80
		$n_{\text{epoch}} = 2, \beta = 0.2, smooth = 0.5$	49.31	55.42
		$n_{\text{epoch}} = 2, \beta = 0.4, smooth = 0.5$	49.62	55.53
		$n_{\text{epoch}} = 2, \beta = 0.6, smooth = 0.5$	48.83	55.14
		$n_{\text{epoch}} = 2, \beta = 0.8, smooth = 0.5$	50.34	54.69

Table 6.1: Results of Different Unsupervised Representation Learning Algorithms on the Contextual Mention Retrieval Task.

objective is helping the VAE extract more useful features from the raw bag-of-words representation.

6.1.2.2 The influence of Decoder Type on VAE

The decoder type choice for the VAE exerts significant influence on the performance of the VAE. The best performance with the multilabel decoder for the VAE is 40.99 which increases by 10 absolute points to 50.91 in the case of the multinomial decoder. This shows us that the multinomial decoder is the better choice for encoding high-dimensional sparse bag-of-words representations of text documents.

6.1.2.3 The effectiveness of VAE training

An interesting observation is the relatively small improvement in the individual performance of the VAE from 50.74% infAP to 50.91% infAP after the multinomial decoder is trained for 5 epochs. However, the improvement in the ensemble-performance is larger from 54.05% to 55.80% which is almost a 2% absolute improvement in inferred average precision.

6.2 Named Entity Disambiguation

The goal of Named Entity Disambiguation (NED) is to link a detected name-mention in a text document to an entity in a knowledge graph (KG). See Hoffart et al. (2011) for an overview of the NED task and survey of approaches circa 2011. See (Bollacker et al. 2008; Dong et al. 2014) for an introduction to knowledge graphs.

More recently, unsupervised representation-learning approaches – such as

Word2Vec (Mikolov et al. 2013), Paragraph Vectors (Le and Mikolov 2014), and BERT (Devlin et al. 2018a) – have become popular for language processing tasks. Simultaneously systems that learn low-dimensional and dense *Entity Embeddings* were proposed for the NED task by He et al. (2013), Yamada et al. (2016), Fang et al. (2016), Zwicklbauer, Seifert, and Granitzer (2016), and Yamada et al. (2017) and Ganea and Hofmann (2017). In light of the greatly increased research into entity-embeddings, and because of the sustained interest in solving the NED task, the investigation – presented in this paper – of the effects of different entity embedding methods on NED accuracy will be useful.

One of the current best NED systems is the document-level joint-inference neural model proposed by (Ganea and Hofmann 2017). This model roughly operates along the following three steps:

In Step One: A max-margin objective is optimized – independently for each entity – to learn an entity embedding from its description page and the tokens in a fixed size window surrounding its mentions. The optimization procedure itself is inspired by Word2Vec and relies on negative sampling. The entity embeddings are learned separately, and then they are frozen.

In Step Two: For each mention, a list of top- k candidate entities are re-scored using a *local* neural network which receives the mention its context and the entity embeddings as input.

Finally, in Step Three: a document level joint-inference procedure⁶ is used to determine which entity is referred by which mention.

Although (Ganea and Hofmann 2017) quantified the downstream impact on

⁶Specifically loopy-belief propagation over a full-connected factor graph with pairwise potentials is used for joint inference.

the NED accuracy of using a *global* joint-inference model in comparison to a *local* NED model, the downstream impact of the entity embedding method in step one on the final NED accuracy remains unclear. Therefore, in this paper, we quantify the effect of different pre-training objectives and different types of input contexts on NED accuracy.

6.2.1 Entity Embedding: Methods and Data

Let \mathcal{E} denote the set of entities. Abstractly an entity embedding is a map, say $\mathbf{e} : \mathcal{E} \rightarrow \mathbb{R}^d$. d is typically chosen to be between 100 to 500 based on cross-validation with the most common choice being 300. Qualitatively, an entity’s embedding should be discriminative amongst homonym entities such as a *(river) bank* and a *(financial) bank*, and it should be similar to the combined representation of content words that co-occur with its mentions.

6.2.1.1 Data Sources

Mainly two sources of data have been used in previous work for learning \mathbf{e} : **The first data-source** is the text in the *canonical* page that describes an entity. For example, the Wikipedia page for ANARCHY defines the concept of anarchy. It mentions that ANARCHY is a type of *political philosophy* which *rejects hierarchy*. Clearly, embedding the content words that appear on the canonical page of an entity into a low-dimensional dense feature vector can help us describe an entity succinctly. We denote this type of data in general as \mathcal{U}_1 .

The second data-source – which may not exist in some cases – consists of tokens surrounding the mentions of an entity. For example, the Wikipedia

page for ALEXANDER GROTHENDIECK, the famous algebraic geometer, mentions that *Grothendieck was born in Berlin to [anarchist](Anarchism) parents ...*⁷. By optimizing the entity embedding for ANARCHY to be similar to the representations of the tokens surrounding the word *anarchist* in this sentence, we can disambiguate the political philosophy of Grothendieck’s parents. We denote this data-source as \mathcal{U}_2 . Take note that this data-source is not truly unlabeled, and it may not exist in many practical applications. Especially in *cold-start* situations (TAC-KBP@NIST 2015; Rastogi, Lyzinski, and Van Durme 2017) large manually hyperlinked entity corpora do not exist. Therefore entity-embedding models that can operate with or without entity mentions are desirable. If \mathcal{U}_2 is available then the total data is $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ otherwise $\mathcal{U} = \mathcal{U}_1$.

In addition, it is useful to quantify the benefit of entity mentions on well-studied NED task such as *wikification* (Mihalcea and Csomai 2007; Ratino et al. 2011) where large quantities of hyperlinked mentions are readily available, so that we can quantify how much may we gain by annotating this information.

6.2.1.2 Methods

Let \mathcal{W} denote the word vocabulary of our entity embedding system. Contemporary approaches for learning entity embeddings (Yamada et al. 2017; Ganea and Hofmann 2017) start with a pre-trained word embedding map $\mathbf{w} : \mathcal{W} \rightarrow \mathbb{R}^d$.⁸ For example, if the word embeddings of *philosophy* and *theory* are similar and the embeddings for *rejects* and *shuns* are similar then an NED system could correctly disambiguate a *political theory that shuns hierarchy*. It is folk-knowledge that

⁷[description](link) is the markdown hyperlink syntax.

⁸Typically the dimensionality of the word embedding map \mathbf{w} and the entity embedding map \mathbf{e} is kept the same to reduce the number of free hyper-parameters and to map words and entities to the same vector space. We follow the same convention in this work.

using a pre-trained embedding such as Word2Vec pre-trained vectors (Mikolov et al. 2013) improves performance, but the downstream impact on NED – to the best of our knowledge – has not been reported in previous work.

6.2.1.2.1 Max-Margin Entity Embedding The word embedding map \mathbf{w} and unlabeled data $\mathcal{U}_{\{1,2\}}$ can be used to learn \mathbf{e} in many ways. However, Word2Vec inspired objectives have dominated other approaches in the context of NED research (Yamada et al. 2016; Ganea and Hofmann 2017; Yamada, Shindo, and Takefuji 2018). In this paper, we focus on a state-of-the-art model for entity disambiguation proposed by (Ganea and Hofmann 2017) which minimizes a max-margin loss for learning entity embeddings.

(Ganea and Hofmann 2017) motivate their objective as follows: Let \mathcal{U}_e denote the portion of training data containing all the words that co-occur with e . Let $p(w|e)$ denote a conditional-multinomial distribution of words that occur in \mathcal{U}_e . $p(w|e)$ is estimated from empirical counts $\#(w, e) / \sum_{w \in \mathcal{U}_e} \#(w, e)$. Next, let $q(w)$ be an unconditional probability distribution. (Ganea and Hofmann 2017) define $q(w) = p(w)^\alpha$ for some $\alpha \in (0, 1)$ where $p(w) \propto \#(w)$ in \mathcal{U} . Let w^+, w^- be two random variables sampled from $p(w|e)$ and $q(w)$ respectively. Careful readers will have noticed that the setup so far is the same as (Mikolov et al. 2013). Indeed, this is why we said that these models are “Word2Vec inspired”.

The novelty of (Ganea and Hofmann 2017) is that they optimize the max-margin objective in (6.1) instead of the logistic-type loss defined by (Mikolov

et al. 2013) to infer the optimal embedding for entity e with a margin hyper-parameter $\gamma > 0$:

$$\begin{aligned}\mathbf{e}(e) &= \arg \min_{\mathbf{z}: \|\mathbf{z}\|=1, \mathbf{z} \in \mathbb{R}^d} \mathbb{E}_{w^+|e} \mathbb{E}_{w^-} \left[h(\mathbf{z}; w^+, w^-) \right] \\ h(\mathbf{z}; w, v) &= \max(0, \gamma - \mathbf{z}^T(\mathbf{w}_w - \mathbf{w}_v))\end{aligned}\tag{6.1}$$

Since the word-embeddings are kept fixed therefore the above objective is convex.

6.2.1.2.2 Variational Entity Embedding There are many ways of constructing the entity embedding map \mathbf{e} from \mathbf{w} and \mathcal{U} . (Ganea and Hofmann 2017) motivated their learning algorithm using generative assumptions but optimized a *contrastive* max-margin objective in the actual learning process. At this step, a question naturally arises that how important is the max-margin algorithm for learning and what other methods could be motivated from the same generative assumption.

In order to answer these questions, we propose to use the Variational-Autoencoder Framework (VAE) (Kingma and Welling 2014b) for learning entity embeddings and comparing their downstream performance to the max-margin entity embeddings by (Ganea and Hofmann 2017). Under the standard VAE framework, the generative model is:

$$\mathbf{z} \sim \pi = \mathcal{N}(\mathbf{0}, \mathbf{I}), \text{ and } \mathbf{x}_e \sim p(w|\mathbf{z}) = \text{NN}_{\boldsymbol{\theta}}^g(\mathbf{z})$$

Here π denotes the standard gaussian prior distribution on the latent variable \mathbf{z} , the output of NN^g are the mean-parameters of a conditional multinomial distribution. \mathbf{x}_e denotes the bag-of-words representation of \mathcal{U}_e .⁹ The parameters

⁹Recall that in the previous section, we defined \mathcal{U}_e as the training words that co-occur with e .

θ of NN^g are learnt by defining two inference-networks $\text{NN}_\phi^{i,m}, \text{NN}_\phi^{i,v}$ that map \mathbf{x}_e to a gaussian posterior over the latent variable \mathbf{z} . $\text{NN}_\phi^{i,m}, \text{NN}_\phi^{i,v}$ compute the mean and variance of the posterior respectively. The parameters θ, ϕ are learnt jointly by minimizing the Evidence Lower Bound Objective (ELBO):

$$\begin{aligned} \arg \min_{\theta, \phi} \sum_{e \in \mathcal{U}} E_{\text{NN}_\phi^i(\mathbf{z}|\mathbf{x}_e)} [\log \text{NN}_\theta^g(\mathbf{x}_e|\mathbf{z})] \\ - \beta \text{KL}(\text{NN}_\phi^i(\mathbf{z}|\mathbf{x}_e) || \pi). \end{aligned} \quad (6.2)$$

Here $\beta \geq 0$ is a hyper-parameter that can be tuned via cross-validation. A larger value of β leads to disentangled representations (Higgins et al. 2017) but a lower value of β can improve the model fit by decreasing the KL penalty. Another interpretation of $\beta = 0$ is that instead of using π as the prior of \mathbf{z} we are using a dynamic prior equal to $\text{NN}_\phi^i(\mathbf{z}|\mathbf{x}_e)$. After the training we define $\mathbf{e}(e)$ as the posterior mean of \mathbf{z} given \mathbf{x}_e , i.e.,

$$\mathbf{e}(e) = \text{NN}_\phi^{i,m}(\mathbf{x}_e). \quad (6.3)$$

6.2.1.2.3 Null Objective Entity Embedding Recall that we defined \mathbf{x}_e as the bag-of-words vector representation of \mathcal{U}_e . The VAE method learns a neural network to map \mathbf{x}_e to $\mathbf{e}(e)$ as shown in (6.3). We want to know how beneficial is the VAE objective itself in training a discriminative neural network. Is it the case that \mathbf{x}_e are themselves linearly separable and a random low dimensional embedding will work just as well or better than the VAE? In order to answer these questions, we conduct an experiment where we use (6.3) with a randomly initialized inference network as our entity embedding.

We call this the *Null Objective* Entity Embedding because we do not optimize objective (6.2) for training $\text{NN}_\phi^{i,m}$.

6.2.2 Related Works

Recently (Kar et al. 2018) also released a local model; however, the code they have released was incomplete and Yamada et al. also released entity embeddings. In order to limit the scope of this study, I restricted myself to the global model proposed by and we use the variational autoencoder to closely mimic the generative assumptions in the original paper but without the max-margin training. Our model is most closely related to the NVDM model (Miao, Yu, and Blunsom 2016).

There are potentially many ways of either bag-of-words methods such as Paragraph Vectors (Le and Mikolov 2014), Simple Embedding (Arora, Liang, and Ma 2017) or even pre-trained sentence encoders such as ELMO (Peters et al. 2018a) and BERT (Devlin et al. 2018a) to construct \mathbf{e} . Most recently (Yamada, Shindo, and Takefuji 2018) proposed a method to train entity embeddings; however they did not show a downstream evaluation of NED accuracy. Therefore, it is not known at this time how effective these embeddings will be for NED. We leave this evaluation for future work.

6.2.3 Experiments and Results

To quantify the effect of the embedding objective on NED accuracy we trained the best performing *global* NED model from (Ganea and Hofmann 2017) with pre-trained entity embeddings learnt using different objectives. For each objective, we varied whether the entity embeddings had access to the mentioned context \mathcal{U}_2 . Following prior work, we evaluated the micro F1 score of the trained models on four of the most popular NED datasets. The four datasets

are AIDA-CoNLL (Hoffart et al. 2011), MSNBC (Cucerzan 2007), AQUAINT (AQUA.) (Milne and Witten 2008) and ACE2004 (ACE04) (Ratinov et al. 2011). We used the preprocessed versions of these datasets released by (Ganea and Hofmann 2017; Guo and Barbosa 2014). The F1 scores and pertinent statistics for the datasets are shown in Table 6.2.

The Max-Margin Embeddings were trained on a February 2014 dump of Wikipedia as follows: Each entity vector was randomly initialized from a mean-0, variance-1, 300-dimensional normal distribution. Values larger than 10 were clipped. First, the embeddings were trained to convergence on the content words in the canonical description page for an entity. In each iteration, 20 samples of w^+ and 5 samples of w^- were used to minimize (6.1). The optimizer was Adagrad with a learning rate of 0.3. Hyperparameters $\alpha = 0.6$ and $\gamma = 0.1$ were the recommended values from (Ganea and Hofmann 2017). If \mathcal{U}_2 was included, then we used the tokens from a window of size 20 as positive examples as well.

The 300-dimensional VAE embeddings were trained using a fixed vocabulary of 50,000 words.

For learning the NED model, we used ADAM with a learning rate of 1e-4 until the validation accuracy exceeded a threshold. Afterward, we reduced the learning rate to 1e-5. The validation threshold was selected for each configuration by first training the system with a threshold of 100% for three trials and then using the median of the highest validation F1 scores. The rest of the training details of the global NED model were identical to (Ganea and Hofmann 2017).

Objective	\mathcal{U}_2	AIDA (4485) (98.2%)	MSNBC (656) (98.5%)	AQUA (727) (94.2%)	ACE04 (257) (90.6%)
Max	✗	89.5	92.7	84.9	88.1
Margin	✓	92.2	93.7	88.5	88.5
VAE	✗	83.9	93.3	84.6	87.7
	✓	85.4	94.7	86.9	86.5
Null	✗	84.1	92.6	81.3	88.1
	✓	83.0	93.2	86.4	88.1

Table 6.2: Micro F1 results for the same NED model but with entity embeddings pre-trained with different objectives. ✓ and ✗ in the second column indicates whether the learning algorithm had access to the mention context or not, respectively. The top two numbers in the dataset columns indicate the test-set size and the recall of the top-30 candidate entities.

6.2.3.1 Entity Relatedness:

In addition to downstream NED evaluation we also evaluated our embeddings on an intrinsic task of entity relatedness prediction. The entity relatedness test set of (Ceccarelli et al. 2013) measures how well the geometry of the entity embeddings captures manually annotated similarity relations between entities. It contains 3319 and 3673 *entity-relatedness queries* for the test and validation sets. Each query consists of one prompt entity and up to one hundred candidate entities. Each candidate has a gold label indicating whether it is related to the prompt entity or not. The cosine similarity of the entity embeddings is used to rank the candidate entities, and the goal is to rank related entities before the unrelated entities. Same as previous work we report Normalized Discounted Cumulative Gain (NDCG) at three different ranks and the Mean Average Precision (MAP) score. Table 6.3 shows the performance of different systems on this test set.

Objective	\mathcal{U}_2	NDCG@1	NDCG@5	NDCG@10	MAP
Max	✗	0.616	0.590	0.616	0.549
Margin	✓	0.646	0.611	0.639	0.576
VAE	✗	0.592	0.559	0.578	0.514
	✓	0.615	0.571	0.596	0.542
Null	✗	0.577	0.537	0.563	0.503
	✓	0.615	0.571	0.596	0.542

Table 6.3: Entity Relatedness Evaluation Results.

6.3 Conclusion

We started with the hypothesis that the entity embeddings based on the variational autoencoder will outperform the MVLSA embeddings and our experiments on the CMR task validated this hypothesis. Moreover, within the different types of decoder types, we found that the multinomial decoder had a significantly higher performance than the multilabel decoder.

We then focused attention on the multinomial VAE generative model and compared its performance to a state-of-the-art contrastive method for learning entity embeddings for the task of named entity disambiguation. We also evaluated the contribution to the NED accuracy by the mention context \mathcal{U}_2 ? The results in Table 6.2 show that although the performance improvement varies with each system and on each dataset, for the best Max-Margin entity embeddings the improvement in performance is substantial.

Based on the experiments we can conclude that although MVLSA and GCCA, in general, are useful methods for learning entity embeddings their utility is limited in situations where multiple views of data are not readily available. On the other hand, variational methods for learning embeddings are promising in single view situations and can fare well in comparison to other more discriminative

methods for learning entity embeddings for NLP tasks.

Chapter 7

Concluding Remarks

In this thesis, I developed novel algorithms for learning representations at the level of words and entities mentioned in linguistic corpora. Chapter 3 focused on multi-view learning of word embeddings using the novel spectral method called MVLSA, and Chapter 4 presented the NVSE model – which builds upon the Variational Auto-Encoder framework – for learning embeddings of entities. Through automatic evaluations for MVLSA and human evaluations for NVSE I showed that these methods can outperform existing state of the art methods in their respective domains.

Then in Chapter 5 I explored ways of geometrically regularizing the learning of entity embeddings in a knowledge base to force the learnt embeddings to comply with logical constraints. I showed that on toy tasks at least it is possible to perform interesting logical inference using the proposed regularization methods. Chapter 6 applied the proposed MVLSA, NVSE algorithms to more practical, downstream tasks of Contextual Mention Retrieval (CMR) and Named Entity disambiguation (NED). We found that an ensemble of the features learnt through the variational-autoencoder approach with pre-existing bag-of-words features

can improve the performance of a state-of-the-art CMR system. However on the task of Named Entity Disambiguation we did not find any benefit of our representations over the state of the art entity embeddings.

Future directions for some of the work in this thesis, especially regarding the NVSE algorithm, involves the use of pretrained contextual sequence encoders such as ELMO (Peters et al. 2018b) and BERT (Devlin et al. 2018b). The methods for extracting features for entities proposed in this thesis should be compatible with these sequence encoding methods. Another potential future direction will be to use the constraint based methods developed in this thesis for embedding entities in knowledge graphs and applying them to more sophisticated models of entities such as the Box-Lattice Measures for probabilistic embeddings (Vilnis et al. 2018; Li et al. 2018) and (Subramanian and Chakrabarti 2018). In order to do so, it will be desirable to improve the scalability of the alternative projection stochastic gradient algorithm proposed in Chapter 5.

Appendix A

A.1 Bayesian Sets

The Bayesian Sets algorithm ranks the elements in $\mathcal{X} \setminus \mathcal{Q}$ according to the ratio of two probabilities:

$$score(x) = \frac{p(x|\mathcal{Q})}{p(x)} = \frac{E_{p(z|\mathcal{Q})}[p(x|z)]}{E_{\pi(z)}[p(x|z)]}$$

Instead of assuming the commonly used Beta-Binomial distribution I assume that $p(x|z)$ is a product of independent Poisson distributions with Gamma conjugate priors. I.e. $p(x|z) = \prod_k \frac{z_k^{x_k}}{x_k!}$. The conjugate prior on z is a product of Gamma distributions,

$$p(z|\alpha, \beta) = \prod_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} z_k^{\alpha_k-1} \exp(-\beta_k z_k)$$

. Let $f(x_k, \alpha_k, \beta_k) =$

$$\left(\frac{x_k + \alpha_k - 1}{x_k} \right) \left(1 - \frac{1}{1 + \beta_k} \right)^{\alpha_k} \left(\frac{1}{1 + \beta_k} \right)^{x_k}.$$

The Bayesian Sest score under these conditions is

$$score(x) = \prod_k \frac{f(x_k, \tilde{\alpha}_k, \tilde{\beta}_k)}{f(x_k, \alpha_k, \beta_k)}$$

Where $\tilde{\alpha}_k = \alpha_k + \sum_{x \in \mathcal{Q}} x_k$ and $\tilde{\beta}_k = \beta_k + Q$. Note that if $\tilde{\alpha}_k = \alpha_k$ then $\frac{f(x_k, \tilde{\alpha}_k, \tilde{\beta}_k)}{f(x_k, \alpha_k, \beta_k)} = \left(\frac{1 + \beta_k}{1 + \beta_k + D} \right)^{x_k}$ which means that features that occur in x that did not

occur in \mathcal{Q} are penalized based on the number of times the feature appeared. Therefore, the Gamma-Poisson distribution is a good approximation only when quantitative differences in the number of times a feature appears are important.

Finally I assume that the components of x were sampled from conditionally independent gaussian distributions with unknown mean and precisions. I.e.

$$p(x|\mu, \tau) =$$

$$\prod_k \sqrt{\frac{\tau}{2\pi}} \exp(-(x_k - \mu_k)^2 \tau_k)$$

and $p(\mu, \tau|\rho, \lambda, \alpha, \beta) =$

$$\prod_k \frac{\beta_k^{\alpha_k} \sqrt{\lambda_k}}{\Gamma(\alpha_k) \sqrt{2\pi}} \tau_k^{\alpha_k - \frac{1}{2}} \exp(-\beta_k \tau_k) \exp(-\frac{\lambda_k \tau_k (\mu_k - \rho_k)^2}{2}).$$

In the following formulae I omit the subscript k for convenience.

$$\bar{x} = \frac{1}{Q} \sum_{x \in \mathcal{Q}} x$$

$$\tilde{\rho} = \frac{\lambda \rho + Q \bar{x}}{\lambda + Q}$$

$$\tilde{\lambda} = \lambda + Q$$

$$\tilde{\alpha} = \alpha + Q/2$$

$$\tilde{\beta} = \beta + \frac{1}{2} \sum_{x \in \mathcal{Q}} (x - \bar{x})^2 + \frac{Q\lambda}{Q + \lambda} \frac{(\bar{x} - \tilde{\rho})^2}{2}$$

The Bayesian Sets score is the ratio of two t distribution values:

$$score(x) = \prod_k \frac{t_{2\tilde{\alpha}_k}(x_k | \tilde{\rho}_k, \frac{\tilde{\beta}_k(\tilde{\lambda}_k+1)}{\tilde{\alpha}_k \tilde{\lambda}_k})}{t_{2\alpha_k}(x_k | \rho_k, \frac{\beta_k(\lambda_k+1)}{\alpha_k \lambda_k})}$$

Now the value of $t_\nu(x|a, b)$ where a is the location parameter and b is the

scale parameter is:

$$t_\nu(x|a, b) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{b\nu\pi}\Gamma(\frac{\nu}{2})} \left(1 + \frac{(x-a)^2}{b\nu}\right)^{-\frac{\nu+1}{2}}$$

In order to use this distribution with count data, it is important to use some variance stabilizing transform, and then perform mean and variance normalization to preprocess all the count features. In this way I can set the priors $\tilde{\rho}_k$ to be 0 and λ_k can be set uniformly to some small number such as 2 and α_k, β_k can be chosen to be 2, 1 respectively.

A.1.1 Binarizing feature counts

BS binarizes the feature vector f_x as f'_x via thresholding:

$$f'_x[j] = \mathbb{I}[f_x[j] > \mu[j] + \lambda\sigma[j]]$$

$$\mu[j] = \frac{\sum_{x \in \mathcal{X}} f_x[j]}{X}, \sigma^2[j] = \frac{\sum_{x \in \mathcal{X}} (f_x[j] - \mu[j])^2}{X},$$

where $\lambda \in \mathbb{R}$ is a hyperparameter. I tried three values of $\lambda - \{0, 0.5, 1\}$ – and set it to 0.5 based on preliminary experiments. BS's scoring function becomes

$$\text{score}_{BS}(\mathcal{Q}, x) = \sum_{j=1}^F \left(\log \frac{\tilde{\alpha}_{\mathcal{Q}}[j]\beta[j]}{\alpha[j]\tilde{\beta}_{\mathcal{Q}}[j]} \right) f'_x[j] \quad (\text{A.1a})$$

$$\tilde{\alpha}_{\mathcal{Q}}[j] = \alpha[j] + \sum_{x \in \mathcal{Q}} f'_x[j] \quad (\text{A.1b})$$

$$\tilde{\beta}_{\mathcal{Q}}[j] = \beta[j] + Q - \sum_{x \in \mathcal{Q}} f'_x[j]. \quad (\text{A.1c})$$

A.2 Ranking methods

A standard function for computing the distance between distributions is the KL-divergence. Another possibility to compute the distance between distributions

is to compute the symmetric version of the KL-divergence. Another standard method for computing the similarity between two probability distributions is to compute the probability product kernel (PPK) between two distributions (Jebara, Kondor, and Howard 2004); i.e.

$$\langle q_\phi(z|\mathcal{Q}), q_\phi(z|x) \rangle = \int_z q_\phi(z|\mathcal{Q}) q_\phi(z|x) dz$$

In the special case that $q_\phi(z|\mathcal{Q})$ and $q_\phi(z|x)$ have the special deep-gaussian form then the KL divergence as well as the inner product can be computed in closed form. KL Divergence between two distributions normal distributions p_1, p_2 with parameters (μ_1, Σ_1) and (μ_2, Σ_2) is:

$$KL(p_1||p_2) = \frac{1}{2} \left(\text{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^\top \Sigma_2^{-1}(\mu_1 - \mu_2) - d + \log \frac{\det(\Sigma_2)}{\det(\Sigma_1)} \right).$$

and PPK is

$$\exp\left(\frac{-(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)}{2} - \log \det((\Sigma_1 + \Sigma_2))\right)$$

In the further special case that $\mu_2 = \mathbf{0}, \Sigma_2 = \mathbf{I}$ then the KL divergence simplifies to:

$$KL(p_1||p_2) = \frac{1}{2} \left(\text{tr}(\Sigma_1) + \mu_1^T \mu_1 - d - \log \det(\Sigma_1) \right).$$

However, I propose here a simple way to compute the distance between two normal distributions. If μ_1, Σ_1 and μ_2, Σ_2 are the mean and variance of two normal distributions, p_1, p_2 then I use the following distance

$$d(p_1, p_2) = \|\mu_1 \Sigma_1^{-1} - \mu_2 \Sigma_2^{-1}\|^2 = \|\xi_1 - \xi_2\|^2$$

This metric can be implemented as a single matrix multiplication while KL divergence and PPK cannot. Intuitively this distance gives higher weightage to those dimensions where the variance of the either the distributions is lower. In

preliminary experiments I found this distance to be superior to KL divergence and PPL and I have used this distance function in all of my experiments. I believe that the regularization from the gaussian prior that encourages the posterior distributions to be close to the origin make shift invariance unnecessary.

References

- Landauer, Thomas K, and Susan T Dumais. 1997. “A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.” *Psychological review* 104 (2): 211. [mvppdb](#).
- Rastogi, Pushpendre, Benjamin Van Durme, and Raman Arora. 2015. “Multiview LSA: Representation Learning Via Generalized CCA”. In *Proceedings of NAACL*.
- Rastogi, Pushpendre, Adam Poliak, Vince Lyzinski, and Benjamin Van Durme. 2018. “Neural variational entity set expansion for automatically populated knowledge graphs”. *Information Retrieval Journal*. ISSN: 1573-7659. doi:[10.1007/s10791-018-9342-1](#). <https://doi.org/10.1007/s10791-018-9342-1>.
- Rastogi, Pushpendre, Adam Poliak, and Benjamin Van Durme. 2017. “Training Relation Embeddings under Logical Constraints”. In *Proceedings of the First Workshop on Knowledge Graphs and Semantics for Text Retrieval and Analysis (KG4IR 2017)*, ed. by Laura Dietz, Chenyan Xiong, and Edgar Meij, 25–31. Tokyo, Japan. http://ceur-ws.org/Vol-1883/paper_9.pdf.
- Nigam, Kamal, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. 1998. “Learning to classify text from labeled and unlabeled documents”. In *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, 792–799. Madison, US: AAAI Press, Menlo Park, US. <http://www.cs.cmu.edu/~knigam/papers/emcat-aaai98.ps>.
- Liu, Ye, Yuxuan Liang, Shuming Liu, and David S. Rosenblum. 2016b. “Urban Water Quality Prediction based on Multi-task Multi-view Learning”. *IJCAI 2016*. <https://www.microsoft.com/en-us/research/publication/urban-water-quality-prediction-based-multi-task-multi-view-learning-2/>.
- He, Jingrui, and Rick Lawrence. 2011. “A Graphbased Framework for Multi-Task Multi-View Learning.” In *ICML*, 25–32.

- Fu, Yanwei, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. 2015. "Transductive multi-view zero-shot learning". *IEEE transactions on pattern analysis and machine intelligence* 37 (11): 2332–2345.
- Tipping, Michael E, and Christopher M Bishop. 1999. "Probabilistic principal component analysis". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61 (3): 611–622.
- Roweis, Sam. 1998. "EM algorithms for PCA and SPCA". *Advances in neural information processing systems*: 626–632.
- Spearman, Charles. 1904. "General Intelligence," objectively determined and measured". *The American Journal of Psychology* 15 (2): 201–292.
- Thurstone, L. L. 1947. *Multiple-factor analysis; a development and expansion of The Vectors of Mind*. xix, 535–xix, 535. Chicago, IL, US: University of Chicago Press.
- Saul, Lawrence K, and Sam T Roweis. 2003. "Think globally, fit locally: unsupervised learning of low dimensional manifolds". *Journal of machine learning research* 4 (06): 119–155.
- Belkin, Mikhail, and Partha Niyogi. 2003. "Laplacian eigenmaps for dimensionality reduction and data representation". *Neural computation* 15 (6): 1373–1396.
- Tenenbaum, Joshua B, Vin De Silva, and John C Langford. 2000. "A global geometric framework for nonlinear dimensionality reduction". *science* 290 (5500): 2319–2323.
- Bengio, Yoshua, Jean-françois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas L Roux, and Marie Ouimet. 2004. "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering". In *Advances in neural information processing systems*, 177–184.
- Hinton, G. E., and R. R. Salakhutdinov. 2006. "Reducing the Dimensionality of Data with Neural Networks". *Science* 313:504–507. doi:[10.1126/science.1127647](https://doi.org/10.1126/science.1127647).
- Hinton, Geoffrey E, Simon Osindero, and Yee-Whye Teh. 2006. "A fast learning algorithm for deep belief nets". *Neural computation* 18 (7): 1527–1554.
- Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. 2007. "Restricted Boltzmann machines for collaborative filtering". In *Proceedings of the 24th international conference on Machine learning*, 791–798. ACM.

- Kingma, Diederik P., and Max Welling. 2014a. “Auto-Encoding Variational Bayes”. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*.
- Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. 2014. “Stochastic backpropagation and approximate inference in deep generative models”. *arXiv preprint arXiv:1401.4082*.
- Kingma, Diederik P, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. “Semi-supervised learning with deep generative models”. In *Advances in Neural Information Processing Systems*, 3581–3589.
- Miao, Yishu, Lei Yu, and Phil Blunsom. 2016. “Neural variational inference for text processing”. In *International Conference on Machine Learning*, 1727–1736.
- Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. “Latent dirichlet allocation”. *Journal of machine Learning research* 3 (Jan): 993–1022.
- Bach, Francis R, and Michael I Jordan. 2005. “A probabilistic interpretation of canonical correlation analysis”. *Technical Report 688, Department of Statistics, University of California, Berkeley*.
- Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press. ISBN: 0262018020, 9780262018029.
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.
- MacKay, David J. C. 2002. *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press. ISBN: 0521642981.
- Ngiam, Jiquan, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. “Multimodal deep learning”. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 689–696.
- Sridharan, Karthik, and Sham M Kakade. 2008. “An Information Theoretic Framework for Multi-view Learning.” In *Proceedings of COLT*.
- Blum, Avrim, and Tom Mitchell. 1998. “Combining labeled and unlabeled data with co-training”. In *Proceedings of COLT*. ACM.
- Sindhwani, Vikas, Partha Niyogi, and Mikhail Belkin. 2005. “A co-regularization approach to semi-supervised learning with multiple views”. In *Proceedings of ICML workshop on learning with multiple views*, 74–79. Citeseer.

- Kakade, Sham M, and Dean P Foster. 2007. "Multi-view regression via canonical correlation analysis". In *Learning Theory*. Springer.
- Wang, Weiran, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. "On deep multi-view representation learning". In *International Conference on Machine Learning*, 1083–1092.
- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. "Glove: global vectors for word representation". In *Proceedings of EMNLP*. ACL.
- Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. "Linguistic Regularities in Continuous Space Word Representations". In *Proceedings of NAACL-HLT*, 746–751.
- Arora, Raman, and Karen Livescu. 2014. "Multi-view learning with supervision for transformed bottleneck features". In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2499–2503. IEEE.
- Chaudhuri, Kamalika, Sham M Kakade, Karen Livescu, and Karthik Sridharan. 2009. "Multi-view clustering via canonical correlation analysis". In *Proceedings of the 26th annual international conference on machine learning*, 129–136. ACM.
- Zhang, Qi, Yang Xiao, Wei Dai, Jingfeng Suo, Congzhi Wang, Jun Shi, and Hairong Zheng. 2016. "Deep learning based classification of breast tumors with shear-wave elastography". *Ultrasonics* 72:150–157.
- Vinokourov, Alexei, Nello Cristianini, and John Shawe-Taylor. 2003. "Inferring a semantic representation of text via cross-language correlation analysis". In *Advances in neural information processing systems*, 1497–1504.
- Cao, Guanqun, Alexandros Iosifidis, Moncef Gabbouj, Vijay Raghavan, and Raju Gottumukkala. 2018. "Deep Multi-view Learning to Rank". *arXiv preprint arXiv:1801.10402*.
- Benton, Adrian, Raman Arora, and Mark Dredze. 2016. "Learning Multiview Embeddings of Twitter Users". In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 14–19. Berlin, Germany: Association for Computational Linguistics. <http://anthology.aclweb.org/P16-2003>.
- Hotelling, H. 1935. "The most predictable criterion." *Journal of Educational Psychology* 26 (2): 139 –142. ISSN: 0022-0663. <http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1935-03006-001&site=ehost-live&scope=site>.

- Hardoon, David R., Sandor Szedmak, and John Shawe-Taylor. 2004. “Canonical Correlation Analysis: An Overview with Application to Learning Methods”. *Neural Computation* 16 (12): 2639–2664. doi:[10.1162/0899766042321814](https://doi.org/10.1162/0899766042321814). eprint: <https://doi.org/10.1162/0899766042321814>. <https://doi.org/10.1162/0899766042321814>.
- Hastie, Trevor, Andreas Buja, and Robert Tibshirani. 1995. “Penalized Discriminant Analysis”. *The Annals of Statistics* 23 (1): 73–102. ISSN: 00905364. <http://www.jstor.org/stable/2242400>.
- Ge, Rong, Chi Jin, Praneeth Netrapalli, Aaron Sidford, et al. 2016. “Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis”. In *International Conference on Machine Learning*, 2741–2750.
- Arora, Raman, Teodor Vanislavov Marinov, Poorya Mianjy, and Nati Srebro. 2017. “Stochastic approximation for canonical correlation analysis”. In *Advances in Neural Information Processing Systems*, 4775–4784.
- Gao, Chao, Dan Garber, Nathan Srebro, Jialei Wang, and Weiran Wang. 2017. “Stochastic canonical correlation analysis”. *arXiv preprint arXiv:1702.06533*.
- Allen-Zhu, Zeyuan, and Yuanzhi Li. 2017. “Doubly accelerated methods for faster CCA and generalized eigendecomposition”. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 98–106. JMLR.org.
- AKAHO, S. 2001. “A kernel method for canonical correlation analysis”. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Springer-Verlag.
- Andrew, Galen, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. “Deep canonical correlation analysis”. In *International conference on machine learning*, 1247–1255.
- Kettenring, Jon R. 1971. “Canonical analysis of several sets of variables”. *Biometrika* 58 (3): 433–451.
- Asendorf, Nicholas A. 2015. “Informative Data Fusion: Beyond Canonical Correlation Analysis”. PhD thesis, University of Michigan, Horace H. Rackham School of Graduate Studies. <http://hdl.handle.net/2027.42/113419>.
- Carroll, J. Douglas. 1968. “Generalization of canonical correlation analysis to three or more sets of variables”. In *Proceedings of APA*, vol. 3.

- Brin, Sergey, and Lawrence Page. 1998. “The anatomy of a large-scale hypertextual Web search engine”. *Computer Networks and ISDN Systems* 30 (1-7): 107–117. ISSN: 0169-7552. doi:[10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X). [http://dx.doi.org/10.1016/S0169-7552\(98\)00110-X](http://dx.doi.org/10.1016/S0169-7552(98)00110-X).
- Yin, Dawei, Chikashi Nobata, Jean-Marc Langlois, Yi Chang, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, and et al. 2016. “Ranking Relevance in Yahoo Search”. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. doi:[10.1145/2939672.2939677](https://doi.org/10.1145/2939672.2939677). <http://dx.doi.org/10.1145/2939672.2939677>.
- Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze. 2010. “Introduction to information retrieval”. *Natural Language Engineering* 16 (1): 100–103.
- Balog, Krisztian. 2012. “Expertise Retrieval”. *Foundations and Trends® in Information Retrieval* 6 (2-3): 127–256. ISSN: 1554-0677. doi:[10.1561/15000000024](https://doi.org/10.1561/15000000024). <http://dx.doi.org/10.1561/15000000024>.
- Schütze, Hinrich, David A. Hull, and Jan O. Pedersen. 1995. “A comparison of classifiers and document representations for the routing problem”. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '95*. doi:[10.1145/215206.215365](https://doi.org/10.1145/215206.215365). <http://dx.doi.org/10.1145/215206.215365>.
- Fishkind, D. E., V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe. 2015. “Vertex nomination schemes for membership prediction”. *Ann. Appl. Stat.* 9 (3): 1510–1532. doi:[10.1214/15-AOAS834](https://doi.org/10.1214/15-AOAS834). <http://dx.doi.org/10.1214/15-AOAS834>.
- Talukdar, Partha Pratim, and Fernando Pereira. 2010. “Experiments in graph-based semi-supervised learning methods for class-instance acquisition”. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1473–1481. Association for Computational Linguistics.
- Metzger, Steffen, Ralf Schenkel, and Marcin Sydow. 2013. “QBEES”. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*. doi:[10.1145/2505515.2507873](https://doi.org/10.1145/2505515.2507873). <http://dx.doi.org/10.1145/2505515.2507873>.
- Tong, Simon, and Jeff Dean. 2008. *System and methods for automatically creating lists*.
- Ghahramani, Zoubin, and Katherine A. Heller. 2005. “Bayesian Sets”. In *NIPS*.

- Letham, Benjamin, Cynthia Rudin, and Katherine A. Heller. 2013. “Growing a list”. *Data Mining and Knowledge Discovery* 27 (3): 372–395. ISSN: 1573-756X. doi:[10.1007/s10618-013-0329-7](https://doi.org/10.1007/s10618-013-0329-7). <http://dx.doi.org/10.1007/s10618-013-0329-7>.
- Wang, R. C., and W. W. Cohen. 2007. “Language-Independent Set Expansion of Named Entities Using the Web”. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 342–350. doi:[10.1109/ICDM.2007.104](https://doi.org/10.1109/ICDM.2007.104).
- . 2008. “Iterative Set Expansion of Named Entities Using the Web”. In *2008 Eighth IEEE International Conference on Data Mining*, 1091–1096. doi:[10.1109/ICDM.2008.145](https://doi.org/10.1109/ICDM.2008.145).
- Nigam, Kamal, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. “Text Classification from Labeled and Unlabeled Documents using EM”. *Machine Learning* 39 (2/3): 103–134. <http://www.cs.cmu.edu/~knigam/papers/emcat-mlj99.ps>.
- Denis, François. 1998. “PAC Learning from Positive Statistical Queries”. *Algorithmic Learning Theory*: 112–126. ISSN: 0302-9743. doi:[10.1007/3-540-49730-7_9](https://doi.org/10.1007/3-540-49730-7_9). http://dx.doi.org/10.1007/3-540-49730-7_9.
- Liu, Bing, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. “Partially Supervised Classification of Text Documents”. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 387–394. ICML ’02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1-55860-873-7. <http://dl.acm.org/citation.cfm?id=645531.656022>.
- Liu, Yang, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016a. “Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention”. arXiv: <http://arxiv.org/abs/1605.09090v1> [cs.CL]. <http://arxiv.org/abs/1605.09090v1>.
- Li, Xiao-Li, Lei Zhang, Bing Liu, and See-Kiong Ng. 2010b. “Distributional Similarity vs. PU Learning for Entity Set Expansion”. In *Proceedings of the ACL 2010 Conference Short Papers*, 359–364. Uppsala, Sweden: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P10-2066>.
- Natarajan, Nagarajan. 2015. “Learning with positive and unlabeled examples”. PhD thesis, UT Austin. <https://repositories.lib.utexas.edu/handle/2152/32826>.

- Rao, Nikhil, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. 2015. “Collaborative Filtering with Graph Information: Consistency and Scalable Methods”. In *Advances in Neural Information Processing Systems 28*, ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 2107–2115. Curran Associates, Inc. <http://papers.nips.cc/paper/5938-collaborative-filtering-with-graph-information-consistency-and-scalable-methods.pdf>.
- Li, Hang. 2014. “Learning to Rank for Information Retrieval and Natural Language Processing, Second Edition”. *Synthesis Lectures on Human Language Technologies* 7 (3): 1–121. ISSN: 1947-4059. doi:[10.2200/s00607ed2v01y201410hlt026](https://doi.org/10.2200/s00607ed2v01y201410hlt026). <http://dx.doi.org/10.2200/S00607ED2V01Y201410HLT026>.
- Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge Press. ISBN: 0-521-86571-9.
- Sussman, Daniel L, Minh Tang, Donniell E Fishkind, and Carey E Priebe. 2012. “A consistent adjacency spectral embedding for stochastic blockmodel graphs”. *Journal of the American Statistical Association* 107 (499): 1119–1128.
- Nickel, M., K. Murphy, V. Tresp, and E. Gabrilovich. 2016. “A Review of Relational Machine Learning for Knowledge Graphs”. *Proceedings of the IEEE* 104 (1): 11–33. ISSN: 0018-9219. doi:[10.1109/JPROC.2015.2483592](https://doi.org/10.1109/JPROC.2015.2483592).
- Balog, K., P. Serdyukov, and A. P. de Vries. 2012. “Overview of the TREC 2011 Entity Track”. In *Proceedings of the Twentieth Text REtrieval Conference (TREC 2011)*. NIST.
- Dalton, Jeffrey, Laura Dietz, and James Allan. 2014. “Entity query feature expansion using knowledge base links”. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 365–374. ACM.
- Balog, Krisztian, Marc Bron, and Maarten De Rijke. 2011. “Query modeling for entity search based on terms, categories, and examples”. *ACM Trans. Inf. Syst.* (New York, NY, USA) 29 (4): 22:1–22:31. ISSN: 1046-8188. doi:[10.1145/2037661.2037667](https://doi.org/10.1145/2037661.2037667). <http://doi.acm.org/10.1145/2037661.2037667>.
- Raghavan, Hema, James Allan, and Andrew McCallum. 2004. “An exploration of entity models, collective classification and relation description”. In *Proceedings of KDD Workshop on Link Analysis and Group Detection*.

- Bast, Hannah, Florian Baurle, Björn Buchhold, and Elmar Haußmann. 2014. “Semantic full-text search with broccoli”. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14*. doi:[10.1145/2600428.2611186](https://doi.org/10.1145/2600428.2611186). <http://dx.doi.org/10.1145/2600428.2611186>.
- Li, Chengkai, Ning Yan, Senjuti B. Roy, Lekhendro Lisham, and Gautam Das. 2010a. “Facetedpedia”. *Proceedings of the 19th international conference on World wide web - WWW '10*. doi:[10.1145/1772690.1772757](https://doi.org/10.1145/1772690.1772757). <http://dx.doi.org/10.1145/1772690.1772757>.
- Li, Xiaonan, Chengkai Li, and Cong Yu. 2012. “Entity-Relationship Queries over Wikipedia”. *ACM Transactions on Intelligent Systems and Technology* 3 (4): 1–20. ISSN: 2157-6904. doi:[10.1145/2337542.2337555](https://doi.org/10.1145/2337542.2337555). <http://dx.doi.org/10.1145/2337542.2337555>.
- Hoffart, Johannes, Dragan Milchevski, and Gerhard Weikum. 2014. “STICS”. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14*. doi:[10.1145/2600428.2611177](https://doi.org/10.1145/2600428.2611177). <http://dx.doi.org/10.1145/2600428.2611177>.
- Ernst, Patrick, Amy Siu, Dragan Milchevski, Johannes Hoffart, and Gerhard Weikum. 2016. “DeepLife: An Entity-aware Search, Analytics and Exploration Platform for Health and Life Sciences”. *ACL 2016*: 19.
- Agrawal, Ankur, S. Sudarshan, Ajitav Sahoo, Adil Anis Sandalwala, and Prashant Jaiswal. 2012. “Entity Ranking and Relationship Queries Using an Extended Graph Model”. In *Proceedings of the 18th International Conference on Management of Data*, 80–91. COMAD '12. Pune, India: Computer Society of India. <http://dl.acm.org/citation.cfm?id=2694443.2694459>.
- Sawant, Uma, and Soumen Chakrabarti. 2013. “Learning joint query interpretation and response ranking”. *Proceedings of the 22nd international conference on World Wide Web - WWW '13*. doi:[10.1145/2488388.2488484](https://doi.org/10.1145/2488388.2488484). <http://dx.doi.org/10.1145/2488388.2488484>.
- Joshi, Mandar, Uma Sawant, and Soumen Chakrabarti. 2014. “Knowledge Graph and Corpus Driven Segmentation and Answer Inference for Telegraphic Entity-seeking Queries”. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1104–1114. Doha, Qatar: Association for Computational Linguistics. <http://www.aclweb.org/anthology/D14-1117>.

- Yahya, Mohamed. 2016. "Question Answering and Query Processing for Extended Knowledge Graphs". PhD thesis, Universität des Saarlandes Saarbrücken.
- Savenkov, Denis, and Eugene Agichtein. 2016. "When a Knowledge Base Is Not Enough". *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*. doi:[10.1145/2911451.2911536](https://doi.org/10.1145/2911451.2911536). <http://dx.doi.org/10.1145/2911451.2911536>.
- Xu, Kun, Yansong Feng, Siva Reddy, Songfang Huang, and Dongyan Zhao. 2016. "Enhancing Freebase Question Answering Using Textual Evidence". *arXiv preprint arXiv:1603.00957*.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. "Distributed representations of words and phrases and their compositionality". In *Proceedings of NIPS*.
- Vilnis, Luke, and Andrew McCallum. 2015. "Word representations via Gaussian embedding". In *ICLR*.
- Rudolph, M. R., F. J. R. Ruiz, S. Mandt, and D. M. Blei. 2016. "Exponential Family Embeddings". *ArXiv e-prints*. arXiv: [1608.00778](https://arxiv.org/abs/1608.00778) [stat.ML].
- He, Shizhu, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. "Learning to Represent Knowledge Graphs with Gaussian Embedding". *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15*. doi:[10.1145/2806416.2806502](https://doi.org/10.1145/2806416.2806502). <http://dx.doi.org/10.1145/2806416.2806502>.
- Winograd, Terry. 1972. "Understanding natural language". *Cognitive psychology* 3 (1): 1–191.
- Turney, Peter D, and Patrick Pantel. 2010. "From frequency to meaning: Vector space models of semantics". *Journal of AI Research* 37 (1).
- Horst, Paul. 1961. "Generalized canonical correlations and their applications to experimental data". *Journal of Clinical Psychology* 17 (4).
- Anandkumar, Animashree, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. 2014. "Tensor Decompositions for Learning Latent Variable Models". *JMLR* 15. <http://jmlr.org/papers/v15/anandkumar14b.html>.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements Of Statistical Learning*. Vol. 2. Springer.
- Arora, Raman, and Karen Livescu. 2012. "Kernel CCA for multi-view learning of acoustic features using articulatory measurements." *MLSLP*.

- Savostyanov, Dmitry. 2014. *Efficient way to find SVD of sum of projection matrices?* MathOverflow. eprint: <http://mathoverflow.net/q/178573>. <http://mathoverflow.net/q/178573>.
- Brand, Matthew. 2002. “Incremental singular value decomposition of uncertain data with missing values”. In *Computer Vision—ECCV 2002*, 707–720. Springer.
- . 2006. “Fast low-rank modifications of the thin singular value decomposition”. *Linear algebra and its applications* 415 (1).
- Van De Velden, Michel, and Tammo HA Bijmolt. 2006. “Generalized canonical correlation analysis of matrices with missing rows: a simulation study”. *Psychometrika* 71 (2).
- Velden, Michel van de, and Yoshio Takane. 2012. “Generalized canonical correlation analysis with missing values”. *Computational Statistics* 27 (3).
- Al-Rfou, Rami, Bryan Perozzi, and Steven Skiena. 2013. “Polyglot: Distributed Word Representations for Multilingual NLP”. In *Proceedings of CoNLL*. ACL.
- Ganitkevitch, Juri, Benjamin Van Durme, and Chris Callison-Burch. 2013. “PPDB: The paraphrase database”. In *Proceedings of NAACL-HLT*.
- Napoles, Courtney, Matthew Gormley, and Benjamin Van Durme. 2012. “Annotated Gigaword”. In *Proceedings of NAACL Workshop: AKBC-WEKEX*.
- Rastogi, Pushpendre, and Benjamin Van Durme. 2014. “Augmenting FrameNet Via PPDB”. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*. ACL. <http://www.aclweb.org/anthology/W14-2901>.
- Habash, Nizar, and Bonnie Dorr. 2003. “CATVAR: A database of categorial variations for English”. In *Proceedings of MT Summit*.
- Minnen, Guido, John Carroll, and Darren Pearce. 2001. “Applied morphological processing of English”. *Natural Language Engineering* 7 (03).
- Bruni, Elia, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. “Distributional semantics in technicolor”. In *Proceedings of ACL*. ACL.
- Luong, Minh-Thang, Richard Socher, and Christopher D. Manning. 2013. “Better Word Representations with Recursive Neural Networks for Morphology”. In *Proceedings of CoNLL*. ACL.

- Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. "Improving Word Representations via Global Context and Multiple Word Prototypes". In *Proceedings of ACL*. ACL. <http://dl.acm.org/citation.cfm?id=2390524.2390645>.
- Hill, Felix, Roi Reichart, and Anna Korhonen. 2014. "SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation". *arXiv preprint arXiv:1408.3456*.
- Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2001. "Placing search in context: The concept revisited". In *Proceedings of WWW*. ACM.
- Radinsky, Kira, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. "A Word at a Time: Computing Word Relatedness Using Temporal Semantic Analysis". In *Proceedings of WWW*. ACM. <http://doi.acm.org/10.1145/1963405.1963455>.
- Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. "A study on similarity and relatedness using distributional and WordNet-based approaches". In *Proceedings of NAACL-HLT*. ACL.
- Rubenstein, Herbert, and John B. Goodenough. 1965. "Contextual Correlates of Synonymy". *Communications of the ACM* 8 (10). <http://doi.acm.org/10.1145/365628.365657>.
- Miller, George A., and Walter G. Charles. 1991. "Contextual correlates of semantic similarity". *Language and Cognitive Processes* 6 (1). doi:[10.1080/01690969108406936](https://doi.org/10.1080/01690969108406936). <http://dx.doi.org/10.1080/01690969108406936>.
- Faruqui, Manaal, Jesse Dodge, Sujay Jauhar, Chris Dyer, Eduard Hovy, and Noah Smith. 2014. "Retrofitting Word Vectors to Semantic Lexicons". In *Proceedings of the deep learning and representation learning workshop, NIPS*.
- Hill, Felix, KyungHyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. 2014. "Not All Neural Embeddings are Born Equal". *arXiv preprint arXiv:1410.0718*.
- Steiger, James H. 1980. "Tests for comparing elements of a correlation matrix." *Psychological Bulletin* 87 (2).
- McNemar, Quinn. 1947. "Note on the sampling error of the difference between correlated proportions or percentages". *Psychometrika* 12 (2).
- Yu, Mo, and Mark Dredze. 2014. "Improving Lexical Embeddings with Semantic Knowledge". In *Proceedings of ACL*. ACL.

- Hill, Felix, and Anna Korhonen. 2014. “Learning Abstract Concept Embeddings from Multi-Modal Data: Since You Probably Can’t See What I Mean”. In *Proceedings of EMNLP*. ACL. <http://www.aclweb.org/anthology/D14-1032>.
- Weston, Jason, Sumit Chopra, and Keith Adams. 2014. “#TagSpace: Semantic Embeddings from Hashtags”. In *Proceedings of EMNLP*. Doha, Qatar: ACL. <http://www.aclweb.org/anthology/D14-1194>.
- Dhillon, Paramveer, Dean Foster, and Lyle Ungar. 2011. “Multi-View Learning of Word Embeddings via CCA”. In *Proceedings of NIPS*.
- Dhillon, Paramveer, Jordan Rodu, Dean P Foster, and Lyle H Ungar. 2012. “Two Step CCA: A new spectral method for estimating vector models of words”. In *Proceedings of ICML*. ACM.
- Collobert, Ronan, and Rémi Lebre. 2013. *Word Embeddings through Hellinger PCA*. Tech. rep. Idiap.
- Ravichandran, Deepak, Patrick Pantel, and Eduard Hovy. 2005. “Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering”. In *Proceedings of ACL*.
- Bhagat, Rahul, and Deepak Ravichandran. 2008. “Large scale acquisition of paraphrases for learning surface patterns”. In *Proceedings of ACL-HLT*.
- Chan, Tsz Ping, Chris Callison-Burch, and Benjamin Van Durme. 2011. “Reranking Bilingually Extracted Paraphrases Using Monolingual Distributional Similarity”. In *Proceedings of EMNLP Workshop: GEMS*.
- Zou, Will, Richard Socher, Daniel Cer, and Christopher Manning. 2013. “Bilingual Word Embeddings for Phrase-Based Machine Translation”. In *Proceedings of EMNLP*. ACL.
- Faruqui, Manaal, and Chris Dyer. 2014. “Improving Vector Space Word Representations Using Multilingual Correlation”. In *Proceedings of EACL*. <http://www.aclweb.org/anthology/E14-1049>.
- Bansal, Mohit, Kevin Gimpel, and Karen Livescu. 2014. “Tailoring Continuous Word Representations for Dependency Parsing”. In *Proceedings of ACL*. ACL.
- Levy, Omer, and Yoav Goldberg. 2014. “Dependency-Based Word Embeddings”. In *Proceedings of ACL*. ACL.
- Sun, Ming, Carey E Priebe, and Minh Tang. 2013. “Generalized canonical correlation analysis for disparate data fusion”. *Pattern Recognition Letters* 34 (2): 194–200.

- Tripathi, A. 2011. *Data Fusion and Matching by Maximizing Statistical Dependencies*. Department of Computer Science, University of Helsinki. <http://books.google.com/books?id=MluZkgEACAAJ>.
- Bannard, Colin, and Chris Callison-Burch. 2005. "Paraphrasing with bilingual parallel corpora". In *Proceedings of ACL*. ACL.
- Vía, Javier, Ignacio Santamaría, and Jesús Pérez. 2007. "A learning algorithm for adaptive canonical correlation analysis of several data sets". *Neural Networks* 20 (1).
- Ganchev, Kuzman, Joao Graca, John Blitzer, and Ben Taskar. 2008. "Multi-View Learning over Structured and Non-Identical Outputs". In *Proceedings of UAI*.
- Yarowsky, David. 1995. "Unsupervised WSD rivaling supervised methods". In *Proceedings of ACL*. ACL.
- Collins, Michael, and Yoram Singer. 1999. "Unsupervised models for named entity classification". In *Proceedings of EMNLP*. ACL.
- Brefeld, Ulf, Thomas Gärtner, Tobias Scheffer, and Stefan Wrobel. 2006. "Efficient co-regularised least squares regression". In *Proceedings of ICML*. ACM.
- Wang, Richard C., Nico Schlaef, William W. Cohen, and Eric Nyberg. 2008. "Automatic Set Expansion for List Question Answering". In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 947–954. Honolulu, Hawaii: Association for Computational Linguistics. <http://www.aclweb.org/anthology/D08-1099>.
- Lang, Joel, and James Henderson. 2013. "Graph-Based Seed Set Expansion for Relation Extraction Using Random Walk Hitting Times". In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 772–776. Atlanta, Georgia: Association for Computational Linguistics. <http://www.aclweb.org/anthology/N13-1094>.
- He, Yifan, and Ralph Grishman. 2015. "ICE: Rapid Information Extraction Customization for NLP Novices". In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 31–35. Denver, Colorado: Association for Computational Linguistics. <http://www.aclweb.org/anthology/N15-3007>.
- Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. "Dbpedia: A nucleus for a web of open data". *The semantic web*: 722–735.

- Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. “Freebase: a collaboratively created graph database for structuring human knowledge”. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. ACM.
- Pujara, Jay, Eriq Augustine, and Lise Getoor. 2017. “Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short”. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1751–1756. Copenhagen, Denmark: Association for Computational Linguistics. <https://www.aclweb.org/anthology/D17-1184>.
- Rastogi, Pushpendre, Vince Lyzinski, and Benjamin Van Durme. 2017. *Vertex Nomination on the Cold Start Knowledge Graph*. Tech. rep. Human Language Technology Center of Excellence.
- Jin, Xiangyu, James French, and Jonathan Michel. 2005. *Query formulation for information retrieval by intelligence analysts*. Tech. rep. Citeseer.
- Gadepally, Vijay N, Kara B Greenfield, William M Campbell, Joseph P Campbell, Albert I Reuther, and Braden J Hancock. 2016. *Recommender Systems for the Department of Defense and the Intelligence Community*. Tech. rep. MIT Lincoln Laboratory Lexington United States.
- Mitra, B., and N. Craswell. 2017. “Neural Models for Information Retrieval”. *ArXiv e-prints*. arXiv: [1705.01509](https://arxiv.org/abs/1705.01509) [cs.IR].
- Al-Badrashiny, Mohamed, Jason Bolton, Arun Tejavsi Chaganty, Kevin Clark, Craig Harman, Lifu Huang, Matthew Lamm, Jinhao Lei, Di Lu, Xiaoman Pan, Ashwin Paranjape, Ellie Pavlick, Haoruo Peng, Peng Qi, Pushpendre Rastogi, Abigail See, Kai Sun, Max Thomas, Chen-Tse Tsai, Hao Wu, Boliang Zhang, Chris Callison-Burch, Claire Cardie, Heng Ji, Christopher Manning, Smaranda Muresan, Owen C. Rambow, Dan Roth, Mark Sammons, and Benjamin Van Durme. 2017. “TinkerBell: Cross-lingual Cold-Start Knowledge Base Construction”. In *Text Analysis Conference (TAC)*.
- Gottipati, Swapna, and Jing Jiang. 2011. “Linking entities to a knowledge base with query expansion”. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 804–813. Association for Computational Linguistics.
- Zaidan, Omar, Jason Eisner, and Christine Piatko. 2007. “Using “annotator rationales” to improve machine learning for text categorization”. In *Human Language Technologies 2007: The Conference of the North American Chapter*

of the Association for Computational Linguistics; Proceedings of the Main Conference, 260–267.

- Lee, Wonsung, Kyungwoo Song, and Il-Chul Moon. 2017. “Augmented Variational Autoencoders for Collaborative Filtering with Auxiliary Information”. In *ACM Conference on Information and Knowledge Management*. 6. doi: 10.475/123 4: ACM.
- Paşca, Marius, and Benjamin Van Durme. 2007. “What You Seek Is What You Get: Extraction of Class Attributes from Query Logs.” In *IJCAI*.
- . 2008. “Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs”. *Proceedings of ACL-08: HLT*: 19–27.
- Pantel, Patrick, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. “Web-Scale Distributional Similarity and Entity Set Expansion”. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 938–947. Singapore: Association for Computational Linguistics. <http://www.aclweb.org/anthology/D/D09/D09-1098>.
- He, Yeye, and Dong Xin. 2011. “Seisa: set expansion by iterative similarity aggregation”. In *Proceedings of the 20th international conference on World wide web*, 427–436. ACM.
- Sadamitsu, Kugatsu, Kuniko Saito, Kenji Imamura, and Genichiro Kikui. 2011a. “Entity Set Expansion using Topic information”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 726–731. Portland, Oregon, USA: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P11-2128>.
- Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. 2017. “Deep Sets”. In *Advances in Neural Information Processing Systems 30*, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 3394–3404. Curran Associates, Inc. <http://papers.nips.cc/paper/6931-deep-sets.pdf>.
- Vartak, Manasi, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. “A Meta-Learning Perspective on Cold-Start Recommendations for Items”. In *Advances in Neural Information Processing Systems 30*, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 6907–6917. Curran Associates, Inc.

<http://papers.nips.cc/paper/7266-a-meta-learning-perspective-on-cold-start-recommendations-for-items.pdf>.

- Zheng, Yuyan, Chuan Shi, Xiaohuan Cao, Xiaoli Li, and Bin Wu. 2017. “Entity Set Expansion with Meta Path in Knowledge Graph”. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 317–329. Springer.
- Wang, Richard C, and William W Cohen. 2009. “Automatic set instance extraction using the web”. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, 441–449. Association for Computational Linguistics.
- Lin, Dekang. 1998. “Automatic retrieval and clustering of similar words”. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, 768–774. Association for Computational Linguistics.
- Demartini, Gianluca, Tereza Iofciu, and Arjen P. De Vries. 2010. “Overview of the INEX 2009 Entity Ranking Track”. In *Proceedings of the Focused Retrieval and Evaluation, and 8th International Conference on Initiative for the Evaluation of XML Retrieval*, 254–264. INEX’09. Brisbane, Australia: Springer-Verlag. ISBN: 3-642-14555-8, 978-3-642-14555-1. <http://dl.acm.org/citation.cfm?id=1881065.1881096>.
- Metzger, Steffen, Ralf Schenkel, and Marcin Sydow. 2014. “Aspect-based similar entity search in semantic knowledge graphs with diversity-awareness and relaxation”. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, 1:60–69. IEEE.
- Zhang, Xiangling, Yueguo Chen, Jun Chen, Xiaoyong Du, Ke Wang, and Ji-Rong Wen. 2017. “Entity Set Expansion via Knowledge Graphs”. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1101–1104. SIGIR ’17. Shinjuku, Tokyo, Japan: ACM. ISBN: 978-1-4503-5022-8. doi:[10.1145/3077136.3080732](https://doi.org/10.1145/3077136.3080732). <http://doi.acm.org/10.1145/3077136.3080732>.
- He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. “Neural Collaborative Filtering”. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182. WWW ’17. Perth, Australia: International World Wide Web Conferences Steering Committee. ISBN: 978-1-4503-4913-0. doi:[10.1145/3038912.3052569](https://doi.org/10.1145/3038912.3052569). <https://doi.org/10.1145/3038912.3052569>.

- Li, Xiaopeng, and James She. 2017. “Collaborative Variational Autoencoder for Recommender Systems”. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 305–314. ACM.
- Sadamitsu, Kugatsu, Kuniko Saito, Kenji Imamura, and Genichiro Kikui. 2011b. “Entity Set Expansion using Topic information”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 726–731. Portland, Oregon, USA: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P11-2128>.
- Robertson, Stephen, and Hugo Zaragoza. 2009. “The Probabilistic Relevance Framework: BM25 and Beyond”. *Found. Trends Inf. Retr.* (Hanover, MA, USA) 3 (4): 333–389. ISSN: 1554-0669. doi:[10.1561/15000000019](https://doi.org/10.1561/15000000019). <http://dx.doi.org/10.1561/15000000019>.
- Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- Ghahramani, Zoubin, and Katherine A Heller. 2006. “Bayesian sets”. In *Advances in neural information processing systems*, 435–442.
- Poliak, Adam, Pushpendre Rastogi, M. Patrick Martin, and Benjamin Van Durme. 2017. “Efficient, Compositional, Order-sensitive n-gram Embeddings”. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 503–508. Valencia, Spain: Association for Computational Linguistics. <http://www.aclweb.org/anthology/E17-2081>.
- Shen, Jiaming, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. “SetExpan: Corpus-Based Set Expansion via Context Feature Selection and Rank Ensemble”. In *Machine Learning and Knowledge Discovery in Databases*, ed. by Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski, 288–304. Cham: Springer International Publishing. ISBN: 978-3-319-71249-9.
- Hinton, G. E. 1999. “Products of experts”. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, vol. 1, 1–6 vol.1. doi:[10.1049/cp:19991075](https://doi.org/10.1049/cp:19991075).
- Bouchacourt, Diane, Ryota Tomioka, and Sebastian Nowozin. 2017. “Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations”. *arXiv preprint arXiv:1705.08841*.

- Jebara, Tony, Risi Kondor, and Andrew Howard. 2004. “Probability product kernels”. *Journal of Machine Learning Research* 5 (Jul): 819–844.
- Řehůřek, Radim, and Petr Sojka. 2010. “Software Framework for Topic Modelling with Large Corpora”. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. <http://is.muni.cz/publication/884893/en>.
- BALOG, K. 2009. “Overview of the TREC 2009 entity track”. *Proc. TREC2009*.
- Altszyler, Edgar, Mariano Sigman, and Diego Fernández Slezak. 2016. “Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database”. *CoRR* abs/1610.01520. arXiv: [1610.01520](http://arxiv.org/abs/1610.01520). <http://arxiv.org/abs/1610.01520>.
- Sønderby, Casper Kaae, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. “Ladder variational autoencoders”. In *Advances in neural information processing systems*, 3738–3746.
- Kipf, Thomas N, and Max Welling. 2017. “Semi-supervised classification with graph convolutional networks”. In *Proceedings of ICLR*.
- Rastogi, Pushpendre, and Benjamin Van Durme. 2017. “Predicting Asymmetric Transitive Relations in Knowledge Bases”. In *Proceedings of the First Workshop on Knowledge Graphs and Semantics for Text Retrieval and Analysis (KG4IR 2017)*, ed. by Laura Dietz, Chenyan Xiong, and Edgar Meij, 1–7. Tokyo, Japan. http://ceur-ws.org/Vol-1883/paper_9.pdf.
- Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel. 2011. “A three-way model for collective learning on multi-relational data”. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 809–816.
- Hoffart, Johannes, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia”. *Artificial Intelligence* 194:28–61.
- Yao, Xuchen, and Benjamin Van Durme. 2014. “Information Extraction over Structured Data: Question Answering with Freebase”. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 956–966. Baltimore, Maryland: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P14-1090>.

- Ritter, Alan, Sam Clark, Mausam, and Oren Etzioni. 2011. “Named Entity Recognition in Tweets: An Experimental Study”. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1524–1534. EMNLP ’11. Edinburgh, United Kingdom: Association for Computational Linguistics. ISBN: 978-1-937284-11-4. <http://dl.acm.org/citation.cfm?id=2145432.2145595>.
- Dong, Xin, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. “Knowledge vault: A web-scale approach to probabilistic knowledge fusion”. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 601–610. ACM.
- Miller, George A. 1995. “WordNet: a lexical database for English”. *Communications of the ACM* 38 (11): 39–41.
- Guha, Ramanathan. 2015. “Towards A Model Theory for Distributed Representations”. In *AAAI Spring Symposium Series*. <http://www.aaai.org/ocs/index.php/SSS/SSS15/paper/view/10220>.
- Toutanova, Kristina, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. “Representing Text for Joint Embedding of Text and Knowledge Bases”. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1499–1509. Lisbon, Portugal: Association for Computational Linguistics. <http://aclweb.org/anthology/D15-1174>.
- Garcez, Artur S. d’Avila, Dov M. Gabbay, and Krysia B. Broda. 2002. *Neural-Symbolic Learning System: Foundations and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 1852335122.
- Hammer, Barbara, and Pascal Hitzler. 2007. *Perspectives of neural-symbolic integration*. Vol. 77. Springer.
- Nickel, Maximilian, Xueyan Jiang, and Volker Tresp. 2014. “Reducing the rank of relational factorization models by including observable patterns”. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 1179–1187. MIT Press.
- Nickel, Maximilian, Lorenzo Rosasco, and Tomaso Poggio. 2016. “Holographic embeddings of knowledge graphs”. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 1955–1961*. AAAI Press.

- Bouchard, Guillaume, Sameer Singh, and Théo Trouillon. 2015. “On approximate reasoning capabilities of low-rank vector spaces”. In *2015 AAAI Spring Symposium Series*.
- Grefenstette, Edward. 2013. “Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors”. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 1–10. Atlanta, Georgia, USA: Association for Computational Linguistics. <http://www.aclweb.org/anthology/S13-1001>.
- Rocktäschel, Tim, Matko Bošnjak, Sameer Singh, and Sebastian Riedel. 2014. “Low-Dimensional Embeddings of Logic”. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, 45–49. Baltimore, MD: Association for Computational Linguistics. <http://www.aclweb.org/anthology/W/W14/W14-2409>.
- Rocktäschel, Tim, Sameer Singh, and Sebastian Riedel. 2015. “Injecting Logical Background Knowledge into Embeddings for Relation Extraction”. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Learning Distributed Word Representations for Natural Logic Reasoning*. 2015.
- Bowman, Samuel R, and Christopher Potts. 2015. “Recursive neural networks can learn logical semantics”. *ACL-IJCNLP 2015*: 12.
- Wang, Quan, Bin Wang, and Li Guo. 2015. “Knowledge Base Completion Using Embeddings and Rules”. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 1859–1865. IJCAI’15. Buenos Aires, Argentina: AAAI Press. ISBN: 978-1-57735-738-4. <http://dl.acm.org/citation.cfm?id=2832415.2832507>.
- Guo, Shu, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. “Semantically Smooth Knowledge Graph Embedding”. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 84–94. Beijing, China: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P15-1009>.
- Roweis, Sam T, and Lawrence K Saul. 2000. “Nonlinear dimensionality reduction by locally linear embedding”. *Science* 290 (5500): 2323–2326.

- Demeester, Thomas, Tim Rocktäschel, and Sebastian Riedel. 2016. “Lifted Rule Injection for Relation Embeddings”. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1389–1399. Austin, Texas: Association for Computational Linguistics. <https://aclweb.org/anthology/D16-1146>.
- Vendrov, Ivan, Jamie Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. “Order-Embeddings of Images and Language”. In *ICLR*.
- Hu, Zhiting, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. “Harnessing Deep Neural Networks with Logic Rules”. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2410–2420. Berlin, Germany: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P16-1228>.
- Wang, William Yang, and William W. Cohen. 2016. “Learning First-Order Logic Embeddings via Matrix Factorization”. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*. New York, NY: AAAI.
- Guo, Shu, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. “Jointly Embedding Knowledge Graphs and Logical Rules”. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 192–202. Austin, Texas: Association for Computational Linguistics. <https://aclweb.org/anthology/D16-1019>.
- Tucker, Ledyard R. 1966. “Some mathematical notes on three-mode factor analysis”. *Psychometrika* 31 (3): 279–311. ISSN: 1860-0980. doi:[10.1007/BF02289464](https://doi.org/10.1007/BF02289464). <http://dx.doi.org/10.1007/BF02289464>.
- Yoon, Hee-Geun, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2016. “A Translation-Based Knowledge Graph Embedding Preserving Logical Property of Relations”. In *Proceedings of NAACL-HLT*, 907–916.
- Singh, Sameer, Tim Rocktäschel, and Sebastian Riedel. 2015. “Towards Combined Matrix and Tensor Factorization for Universal Schema Relation Extraction”. In *Workshop on Vector Space Modeling for NLP*, 135–142. Denver, Colorado: ACL. doi:[10.3115/v1/W15-1519](https://doi.org/10.3115/v1/W15-1519). <http://aclweb.org/anthology/W15-1519>.
- Grinberg, Darij. 2015. *Existence and characterization of transitive matrices?* MathOverflow. eprint: <http://mathoverflow.net/q/212808>. <http://mathoverflow.net/q/212808>.

- Nickel, Maximilian, and Douwe Kiela. 2017. “Poincaré Embeddings for Learning Hierarchical Representations”. *arXiv preprint arXiv:1705.08039*.
- Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. “BPR: Bayesian Personalized Ranking from Implicit Feedback”. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461. UAI '09. Montreal, Quebec, Canada: AUAI Press. ISBN: 978-0-9749039-5-8. <http://dl.acm.org/citation.cfm?id=1795114.1795167>.
- Riedel, Sebastian, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. “Relation Extraction with Matrix Factorization and Universal Schemas”. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 74–84. Atlanta, Georgia: Association for Computational Linguistics. <http://www.aclweb.org/anthology/N13-1008>.
- Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. “Translating embeddings for modeling multi-relational data”. In *Advances in Neural Information Processing Systems*, 2787–2795.
- Gruber, P. M. 2007. *Convex and Discrete Geometry*. Springer.
- Russell, Stuart, Peter Norvig, and Artificial Intelligence. 1995. “A modern approach”. *Artificial Intelligence. Prentice-Hall, Englewood Cliffs* 25:27.
- Wang, Zhen, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. “Knowledge graph embedding by translating on hyperplanes”. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119. AAAI Press.
- Lin, Yankai, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. “Learning Entity and Relation Embeddings for Knowledge Graph Completion.” In *The 29th AAAI Conference on Artificial Intelligence*, 2181–2187. AAAI.
- Yang, Bishan, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. “Embedding Entities and Relations for Learning and Inference in Knowledge Bases”. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*. <http://research.microsoft.com/apps/pubs/default.aspx?id=241703>.

- Bagga, Amit, and Breck Baldwin. 1998. “Entity-based cross-document coreferencing using the vector space model”. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, 79–85. Association for Computational Linguistics.
- Mayfield, James, David Alexander, Bonnie J Dorr, Jason Eisner, Tamer Elsayed, Tim Finin, Clayton Fink, Marjorie Freedman, Nikesh Garera, Paul McNamee, et al. 2009. “Cross-Document Coreference Resolution: A Key Technology for Learning by Reading.” In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 9:65–70.
- Sankepally, Rashmi, Tongfei Chen, Benjamin Van Durme, and Douglas W. Oard. 2018. “A Test Collection for Coreferent Mention Retrieval”. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1209–1212. SIGIR ’18. Ann Arbor, MI, USA: ACM. ISBN: 978-1-4503-5657-2. doi:[10.1145/3209978.3210139](https://doi.org/10.1145/3209978.3210139). <http://doi.acm.org/10.1145/3209978.3210139>.
- The ACE 2005 (ACE05) Evaluation Plan*. 2005. Tech. rep. NIST.
- Bentivogli, Luisa, Peter Clark, Ido Dagan, Hoa T. Dang, and Danilo Giampiccolo. 2010. “The sixth PASCAL recognizing textual entailment challenge”. In *Proceedings of TAC*.
- Ji, Heng, Joel Nothman, and Ben Hachey. 2014. “Overview of TAC-KBP2014 Entity Discovery and Linking Tasks”. *Proceedings of Text Analysis Conference (TAC)*: 1333–1339.
- Yilmaz, Emine, and Javed A. Aslam. 2006. “Estimating Average Precision with Incomplete and Imperfect Judgments”. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, 102–111. CIKM ’06. Arlington, Virginia, USA: ACM. ISBN: 1-59593-433-2. doi:[10.1145/1183614.1183633](https://doi.org/10.1145/1183614.1183633). <http://doi.acm.org/10.1145/1183614.1183633>.
- McCandless, Michael, Erik Hatcher, and Otis Gospodnetic. 2010. *Lucene in action: covers Apache Lucene 3.0*. Manning Publications Co.
- Davis, Mark. 2011. *Unicode Standard Annex #29: Unicode Text Segmentation*. Tech. rep. Unicode.org. www.unicode.org/reports/tr29.

- Hoffart, Johannes, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenaу, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. “Robust Disambiguation of Named Entities in Text”. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 782–792. Edinburgh, Scotland, UK.: Association for Computational Linguistics. <http://www.aclweb.org/anthology/D11-1072>.
- Le, Quoc, and Tomas Mikolov. 2014. “Distributed Representations of Sentences and Documents”. In *Proceedings of The 31st International Conference on Machine Learning*, 1188–1196.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. “Bert: Pre-training of deep bidirectional transformers for language understanding”. *arXiv preprint arXiv:1810.04805*.
- He, Zhengyan, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. “Learning Entity Representation for Entity Disambiguation”. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 30–34. Sofia, Bulgaria: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P13-2006>.
- Yamada, Ikuya, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. “Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation”. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 250–259. Berlin, Germany: Association for Computational Linguistics. <http://www.aclweb.org/anthology/K16-1025>.
- Fang, Wei, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. 2016. “Entity Disambiguation by Knowledge and Text Jointly Embedding”. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 260–269. Berlin, Germany: Association for Computational Linguistics. <http://www.aclweb.org/anthology/K16-1026>.
- Zwiclkbauer, Stefan, Christin Seifert, and Michael Granitzer. 2016. “Robust and Collective Entity Disambiguation Through Semantic Embeddings”. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 425–434. SIGIR ’16. Pisa, Italy: ACM. ISBN: 978-1-4503-4069-4. doi:[10.1145/2911451.2911535](https://doi.org/10.1145/2911451.2911535). <http://doi.acm.org/10.1145/2911451.2911535>.

- Yamada, Ikuya, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. “Learning Distributed Representations of Texts and Entities from Knowledge Base”. *Transactions of the Association for Computational Linguistics* 5:397–411. ISSN: 2307-387X. <https://transacl.org/ojs/index.php/tac1/article/view/1065>.
- Ganea, Octavian-Eugen, and Thomas Hofmann. 2017. “Deep Joint Entity Disambiguation with Local Neural Attention”. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2619–2629. Copenhagen, Denmark: Association for Computational Linguistics. <https://www.aclweb.org/anthology/D17-1277>.
- TAC-KBP@NIST. 2015. *Cold Start Knowledge Base Population at TAC 2015 Task Description*. 1.1. NIST.
- Mihalcea, Rada, and Andras Csomai. 2007. “Wikify!: Linking Documents to Encyclopedic Knowledge”. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, 233–242. CIKM ’07. Lisbon, Portugal: ACM. ISBN: 978-1-59593-803-9. doi:[10.1145/1321440.1321475](https://doi.org/10.1145/1321440.1321475). <http://doi.acm.org/10.1145/1321440.1321475>.
- Ratinov, Lev, Dan Roth, Doug Downey, and Mike Anderson. 2011. “Local and Global Algorithms for Disambiguation to Wikipedia”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1375–1384. Portland, Oregon, USA: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P11-1138>.
- Yamada, Ikuya, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018. “Representation Learning of Entities and Documents from Knowledge Base Descriptions”. In *Proceedings of the 27th International Conference on Computational Linguistics*, 190–201. Santa Fe, New Mexico, USA: Association for Computational Linguistics. <http://www.aclweb.org/anthology/C18-1016>.
- Kingma, Diederik P., and Max Welling. 2014b. “Auto-Encoding Variational Bayes”. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*.
- Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In *Proceedings of the Fifth International Conference on Learning Representations (ICLR 2017)*.

- Kar, Rijula, Susmija Reddy, Sourangshu Bhattacharya, Anirban Dasgupta, and Soumen Chakrabarti. 2018. “Task-Specific Representation Learning for Web-Scale Entity Disambiguation”. <https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17281>.
- Arora, Sanjeev, Yingyu Liang, and Tengyu Ma. 2017. “A Simple but Tough-to-Beat Baseline for Sentence Embeddings”. In *Proceedings of the ICLR*. <https://openreview.net/forum?id=SyK00v5xx>.
- Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. “Deep Contextualized Word Representations”. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics. <http://www.aclweb.org/anthology/N18-1202>.
- Cucerzan, Silviu. 2007. “Large-Scale Named Entity Disambiguation Based on Wikipedia Data”. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 708–716. Prague, Czech Republic: Association for Computational Linguistics. <http://www.aclweb.org/anthology/D/D07/D07-1074>.
- Milne, David, and Ian H. Witten. 2008. “Learning to Link with Wikipedia”. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 509–518. CIKM ’08. Napa Valley, California, USA: ACM. ISBN: 978-1-59593-991-3. doi:[10.1145/1458082.1458150](https://doi.org/10.1145/1458082.1458150). <http://doi.acm.org/10.1145/1458082.1458150>.
- Guo, Zhaochen, and Denilson Barbosa. 2014. “Robust Entity Linking via Random Walks”. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 499–508. CIKM ’14. Shanghai, China: ACM. ISBN: 978-1-4503-2598-1. doi:[10.1145/2661829.2661887](https://doi.org/10.1145/2661829.2661887). <http://doi.acm.org/10.1145/2661829.2661887>.
- Ceccarelli, Diego, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. “Learning relatedness measures for entity linking”. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 139–148. ACM.

- Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018b. “Deep Contextualized Word Representations”. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics. doi:[10.18653/v1/N18-1202](https://doi.org/10.18653/v1/N18-1202). <http://www.aclweb.org/anthology/N18-1202>.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. “Bert: Pre-training of deep bidirectional transformers for language understanding”. *arXiv preprint arXiv:1810.04805*.
- Vilnis, Luke, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. “Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures”. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 263–272. Melbourne, Australia: Association for Computational Linguistics. <http://www.aclweb.org/anthology/P18-1025>.
- Li, Xiang, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. 2018. “Smoothing the Geometry of Probabilistic Box Embeddings”. In *International Conference on Learning Representations (ICLR)*, 1037–1040.
- Subramanian, Sandeep, and Soumen Chakrabarti. 2018. “New Embedded Representations and Evaluation Protocols for Inferring Transitive Relations”. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1037–1040. ACM.

Curriculum Vita

Pushpendre Rastogi was born in New Delhi, India on 23 March 1989. He graduated from IIT, Delhi with a bachelors in Electrical Engineering and a Masters in Information and Communication Technology in 2011. His masters' thesis was on the *Stationarity Condition for Fractional Sampling Filters*, advised by Professor Brejesh Lall. He did one summer internship at Mechatronics Private Limited, Pune in 2009, where he worked on FPGA circuit design, and one internship at the Polytechnic University of Hong Kong with Professor Ajay Pathak in 2010, where he worked on algorithms for biometrics. During 2011-12 he worked in Goldman Sachs as an Operations Strategist where he implemented a Fat-Finger alert system. He worked at Aspiring Minds Pvt. Ltd. as an applied researcher in 2012-13, where he worked on Automated English Essay Grading.

In 2013 he started pursuing a Computer Science Ph.D. degree at the Johns Hopkins University under the supervision of Benjamin Van Durme. During his Ph.D. he was a teaching assistant for Courses on Representation Learning and Machine Learning for three semesters and he won the George M.L. Sommerman Engineering Graduate Teaching Assistant Award in 2017. He also interned at Samsung in 2017 and at Amazon in 2018 during his studies. He will join the Alexa team in Amazon as an applied research scientist in 2019.