Exponential Word Embeddings: Models and Approximate Learning

Kekec, Taygun

DOI
10.4233/uuid:3f5e34e1-fb18-42d1-b077-38a1a691a301

Publication date
2019

EXPONENTIAL WORD EMBEDDINGS: MODELS AND APPROXIMATE LEARNING

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof dr. ir. T.J.J.J. van der Hagen chair of the
Board for Doctorates
to be defended publicly on
Friday 7, June 2019 at 12:30 o'clock

by

Taygun KEKEÇ

Master of Science in Mechatronics Engineering,
Sabanci University, Turkey
Born in Istanbul, Turkey

This dissertation has been approved by

promotor: Prof. dr. ir. M. J. T. Reinders
copromotor: Dr. D. M. J. Tax

Composition of the doctoral committee:

Rector Magnificus	
Prof.dr.ir. M. J. T. Reinders,	Delft University of Technology, promotor
Dr. D. M. J. Tax,	Delft University of Technology, copromotor
Independent members:	
Prof.dr. M. A. Larson	Delft University of Technology
Prof.dr. M. Orozco-Alzate	Universidad Nacional de Colombia
Prof.dr. B. van Ginneken	Radboud University Nijmegen
Prof.dr.ir. A. P. de Vries	Radboud University Nijmegen
Other members:	
Dr. G. Bouma	University of Groningen
Reserve members:	
Prof.dr. C.M. Jonker	Delft University of Technology

ISBN 978-94-6366-172-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

This thesis is dedicated to my mother and father;
Bilgen Kekeç and İlhami Kekeç

CONTENTS

1 Introduction **1**
- 1.1 Problem of word meanings . . . 1
- 1.2 The Motivation for Numerical Representations . . . 2
  - 1.2.1 Exponential Word Embeddings . . . 3
  - 1.2.2 Topic Models . . . 5
  - 1.2.3 Markov Random Fields . . . 7
- 1.3 Exponential Family Representations . . . 8
- 1.4 Focus of this thesis . . . 9

2 PAWE: Polysemy Aware Word Embeddings **13**
- 2.1 Abstract . . . 13
- 2.2 Introduction . . . 14
  - 2.2.1 Related Work . . . 14
- 2.3 Distributed Word Embeddings . . . 16
  - 2.3.1 Problem Formulation . . . 16
  - 2.3.2 PAWE Embedding Model . . . 17
  - 2.3.3 Learning . . . 18
- 2.4 Experimental Results . . . 19
  - 2.4.1 Nearest Neighbors Analysis. . . . 20
  - 2.4.2 Word Similarity . . . 21
- 2.5 Conclusions . . . 25

3 Robust Gram Embeddings **27**
- 3.1 Abstract . . . 27
- 3.2 Introduction . . . 28
- 3.3 Robust Gram Embeddings . . . 28
- 3.4 Experiments . . . 30
  - 3.4.1 Model Selection . . . 30
  - 3.4.2 Sensitivity Analysis . . . 31
  - 3.4.3 Word Similarity Performance . . . 32
- 3.5 Conclusion . . . 34

4 Semantic Vector Specializations with Bidirectional Constraint Propagations **35**
- 4.1 Abstract . . . 35
- 4.2 Introduction . . . 36
- 4.3 Proposed Approach . . . 37
  - 4.3.1 Word Vector Models . . . 37
  - 4.3.2 Semantic Word Vector Specializations . . . 38
  - 4.3.3 Bidirectional Constraint Propagations . . . 40
  - 4.3.4 Learning by Controlled Negative Sampling . . . 40

4.4 Experimental Results . . . 41
4.4.1 Quantitative Results . . . 42
4.4.2 Model Selection . . . 43
4.4.3 Embedding Stability . . . 44
4.4.4 Word Similarity Measurements . . . 45
4.5 Conclusion . . . 49

5 Boosted Negative Sampling by Quadratically Constrained Entropy Maximization 51
5.1 Abstract . . . 51
5.2 Introduction . . . 52
5.3 Quadratically Constrained Entropy Maximization . . . 53
5.4 Experiments . . . 57
5.4.1 Exponential Family Density Estimation . . . 57
5.4.2 Word Embeddings Similarity . . . 59
5.4.3 Real world text classification . . . 63
5.5 Conclusions . . . 63

6 Constrain Global Sample Local: Faster and Variance Reduced Word Embeddings 65
6.1 Abstract . . . 65
6.2 Introduction . . . 66
6.3 Constrain Global Sample Local Method . . . 67
6.3.1 Sampling Approximation Gap . . . 67
6.3.2 Global Bands for Approximation Gap . . . 68
6.3.3 Local Context Relevance via Concreteness . . . 68
6.3.4 Locally Relevant Sampling Model . . . 70
6.4 Experiments . . . 71
6.4.1 Performance on Word Similarity . . . 73
6.4.2 Variance Reduction . . . 73
6.4.3 Convergence Rates . . . 74
6.5 Conclusions and Discussions . . . 75

7 Markov Random Suitability Field for Wind Farm Planning 77
7.1 Abstract . . . 77
7.2 Introduction . . . 78
7.3 Modeling wind farm suitability . . . 78
7.3.1 A grid-based model on the two-dimensional Cartesian plane . . . 78
7.3.2 Quantifying the elementary criteria for wind farms . . . 79
7.3.3 Multiple-criteria decision analysis of wind farms . . . 80
7.4 Spatial Suitability Modeling with Markov Random Field . . . 80
7.5 Case Study . . . 83
7.5.1 A grid-based model of Turkey . . . 83
7.5.2 Quantifying the wind farm potential in Turkey . . . 84
7.5.3 Spatially-aware suitability for wind farms in Turkey . . . 85
7.6 Conclusion . . . 87

8 Conclusion and Discussions **89**
8.1 Future Research . . . 91

References **93**

A Appendix A **107**
A.1 Variational Bayes for LDA . . . 107
A.2 Lower Bound . . . 108
A.2.1 γ Variational Update . . . 109
A.2.2 ϕ Variational Update . . . 109

B Appendix B **111**
B.1 Negative Sampling Objective . . . 111
B.2 Smoothing the distribution . . . 112
B.3 Powering the distribution . . . 112

Summary **115**

Samenvatting **117**

Acknowledgements **119**

Curriculum Vitæ **121**

1

INTRODUCTION

1.1. PROBLEM OF WORD MEANINGS

Words are powerful entities. It is a question of interest whether we are usually aware of their broad impact. During the course of history, humankind has identified the potential of words and developed literature in order to describe how powerful words can be. An example is the book of One Thousand and One Nights in which Scheherazade tells a story to the king every night in order to delay her execution. She uses words to save her life.

The power of words lies in their meanings. The problem of how words acquire their meanings has been studied from many perspectives; semantics, philosophy of language, philosophy of mind, linguistics etc. Here, we highlight a few reasons why addressing the meaning of words directly is so challenging:

First, words are imprecise by their nature [1]. In many contexts, it is quite challenging to obtain a precise definition. Even carefully constructed lexical dictionary sources exhibit vagueness despite the fact that they are prepared by a committee of field experts. As a result, there are significant variations in word definitions. Secondly, the meaning of a word is a function of collective decisions. Humans signify their ideas with words, and the inherent meaning of a particular word, or a concept, can change by changes in collective usage in the society [2]. Lastly, words are influenced by the dynamics of a society. Complex historical and social processes drive both word meanings and language grammar to different states [3]. Effective discovery of the cause-effect relations, for explaining what words do mean and how that varies, requires complex workflows. Scientific analysis of such phenomenons are bound to the interaction of multiple disciplines.

In this thesis, we circumvent the grand problem of how word meanings arise and what words do actually mean. We instead aim to utilize computational tools in order to find whether we can contribute by developing some numerical word representations. In this manner, we wish to represent the words on computers, such that similarities between words are accurately learned under time and resource constraints. These numerical word representations can be used as building blocks for natural language processing

1

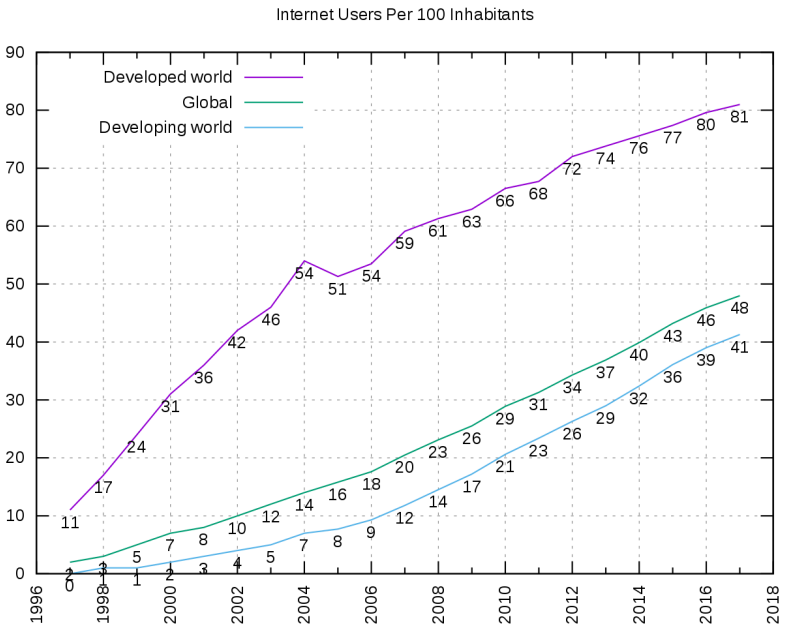


Figure 1.1: Global internet usage statistics from 1996 to 2018 [4]

tasks to address the current needs of the information age, in which we elaborate in the next subsection.

1.2. THE MOTIVATION FOR NUMERICAL REPRESENTATIONS

In the digital age we are living in, we not only see a tremendous improvement both in physical limits of the computation but also on the methods and the speed at which we are able to communicate.

Considering the steady improvements on low cost data storage devices, our overall capacity to collect, filter, process, create and finally distribute more and more information has leveraged. Our price to pay for the accumulation of the increasing amount of digital data is information overload. As we are not able to cope with all the information, we need to develop tools that enable us to combat this information overload and give support to make decisions based on all available information. The information overload necessitates summarizing documents. Out of a large document collection, which subset of documents are more representative? Which parts of the data are more informative? Given the number of documents and the rate at which this number increases, we can not realize these objectives with human labor, and need computational tools to do so.

The scale at which communication takes place has undergone a revolution. In the past, communication and cultural exchange between different societies were taking place on common trade routes and moments of war, whereas interactions between people were almost negligible. Letters were the traditional transmission medium which had often delivery times of weeks. With the invention of telegraphy, it became possible to send messages in minutes of time to thousands of kilometers away. The worldwide internet and its widespread adoption (Fig. 1.1) completely changed our way of communication;

any human on the planet can virtually talk to each other instantly and lengthy.

The way that humans move also radically changed, i.e. lengthy intercontinental journeys over the land have been replaced with daily flights, where previously a subpopulation came into contact with other societies, nowadays everyone has a chance to do so. As a result, more people get into contact with other languages, increasing the need for language translation tools.

We touched upon global trends and their increasing needs of document summarization, and automatic machine translation systems. There are many other Natural Language Processing (NLP) tasks such as Part Of Speech (POS) tagging [5], question answering [6], sentiment analysis [7]. All these tasks require having some form of numerical word representation as the key building block.

Word representations have found their applications in numerous scientific fields. For instance, computer vision benefits from them for text recognition in images [8], image captioning [9] and action recognition [10]. Methodologies are developed for bioinformatics [11] and genomics [12] that uses word representations as side information. All of these systems have potential to benefit from accurate word representations.

In this thesis, we aim to develop a task-independent way of representing words. We do so by learning these representations from examples. As text data usually is ambiguous and noisy, we adopt a probabilistic methodology.

1.2.1. EXPONENTIAL WORD EMBEDDINGS

A word embedding is a representation for each word (symbol) in a language, which we typically learn from long sequence of already preprocessed text. For example, a naive embedding is to use a one-hot encoding, which means that we treat each word as a basis vector $\mathbf{w}_q$ in the vector space with the length of the vector is equal to the vocabulary size. As words are viewed as independent vectors, computing the similarities for words is not possible with this representation. An alternative is the exponential word embeddings which is a recent methodology [13] for obtaining word representations. These representations learn the word vectors with a distributional assumption.

Since there is an immense number of possible configurations these vectors can exhibit, our aim is to learn a representation such that the words with same meanings are similarly represented in the final vector space. Compared to the image representation learning, this is an even more challenging task because:

- Spatial coherence is an acceptable assumption for images. Unfortunately, words usually do not such a continuity property. Take for example: *casting* and *fasting*. A change of one letter already fully changes the meaning of the two words.

- Words are highly abstract symbols. It is difficult to find a connection between their form and meanings.

- An image representation can use information from a vast number of observations. Nowadays, a single image acquired with an ordinary off-the-shelf camera has millions of pixels. The number of observations for a word is however quite a few. Google Books NGram corpus shows that the average word length in English is 5.1 letters [14].

walking in fog covered forest hoping to see a blue sky
. falling tree leaves in forest wonderful scenery for ...
.. losing track in country forest is unlikely with a scout ....
. some thugs sadly started forest fire but buckets of water ...
....scary atmosphere of the forest and storyteller's inspiration
nature gave Ithaca this forest and few beautiful lakes..

Figure 1.2: Illustration of distributional hypothesis for the word *forest*. We generate few sentences containing the *forest* as middle word and. There are semantic relations between *forest*, and co-occurring blue words.

- Semantic and syntactic similarity of words seems to be hard to measure.

DISTRIBUTIONAL HYPOTHESIS

The word embedding approaches we developed in this thesis grounds on the Distributional Hypothesis to represent the similarity of words. This hypothesis was derived from the semantic theory of language usage. The underlying idea of Distributional Hypothesis given in [15] is that: You shall know a word by the company it keeps. Thus, if two words are occurring in the same context, they tend to be similar in their meanings. In Figure 1.2, we provide an illustration of this hypothesis. Here, the word of interest is *forest*, and it occurs more often with words like *tree*, *leaves*, *sky*, suggestive of a semantic relationship between all these words.

The formalization of what Distributional Hypothesis means by context is still an unsolved language processing problem. There are many questions unanswered like: is it practical to take the order of context words into account or rather omit it? Should we use a bilateral context or one-sided context of words? In addition, a theoretical explanation is lacking at how the context length should be chosen. For those interested in different word context implementations, we refer to [16], where the authors provide an excellent literature overview. We now detail neural and matrix based word embedding architectures and explain how they implement the Distributional Hypothesis.

Neural network based embedding architectures implement Distributional Hypothesis by iterating on each training sample, being a sentence of the training set. Lets define $\mathbf{C}_q$ as the set of context words. The conditional probability for a word embedding then becomes the following exponential family model:

$$P(\mathbf{w}_q|\mathbf{C}_q) = \frac{\exp(s_{\boldsymbol{\theta}}(\mathbf{w}_q, \mathbf{C}_q))}{\sum_{\mathbf{w}_{\tilde{q}}} \exp(s_{\boldsymbol{\theta}}(\mathbf{w}_{\tilde{q}}, \mathbf{C}_q))}, \tag{1.1}$$

where $\mathbf{w}_{\tilde{q}}$ iterates over all possible words in the language, and $s_{\boldsymbol{\theta}}$ is the function which decides the similarity between a word and a given context, parameterized by the embedding parameters $\boldsymbol{\theta}$.

Note that two key decisions have to be made in this formulation. Firstly, the form of $\mathbf{C}_q$ has to be determined in the neural architecture. Secondly, the calculation of the denominator can be expensive. There exists a vast amount of work on sampling techniques (e.g. negative sampling technique) to circumvent the calculation of the denominator.

Neural architecture based embeddings update parameters on a sentence basis and they are local models. However, this locality can reduce the learning efficiency if the

training set does not have the right curriculum [17, 18]. In contrast, matrix based embeddings are global alternatives to neural architectures. In matrix based embeddings, the word co-occurrence matrix is first calculated from the given corpus. Then this matrix is decomposed using a Singular Value Decomposition (SVD) to find word vectors such that words that tend to co-occur will be represented by the same eigenvectors. In that sense, they comply with the distribution hypothesis, which is also elaborated in [19].

In some scenarios, we would like to represent documents. In this case, learned word representations allow us to design more sophisticated document representations by incorporating an intermediate function layer. This intermediate layer makes it possible to have a distinction between words and higher level features. It can capture more information such as the style, and mood of the author. In the word embedding literature, such a function that bridges the gap between word vectors and the document representation is called the composition function.

The simplest baseline composition function here is the average word vectors [20]. Although the naive choice of using the average of word vectors to represent a document is a simple technique, we can learn the composition function by fixing word embeddings to get document vectors [21]. Arora et al. proposed a weighted averaging approach followed by a PCA based reduction [22]. In Lee et al. [23], they propose an embedding model called Doc2Vec, extending word vector learning to the paragraph and document vectors. The work of [24] also extends the paragraph vector methodology to a probabilistic fully Bayesian framework rather than obtaining point estimates of paragraph vectors. Here our goal is not going into detail of the composition functions, but pose that word representations can be easily extended using a composition function. We direct the reader to the work of Hill et al. which provides a systematic performance benchmark of document representations that uses a combination of word representations and composition functions [25]. To yield more generality, we adopted average word vector compositions during the thesis.

1.2.2. TOPIC MODELS

Topic modelling is a widely adopted technique for obtaining document representations in NLP. These probabilistic models are based on the key idea that there exist high-level concepts, called topics, that can explain how documents are formed. The number of topics is usually orders of magnitude smaller than the vocabulary size. Thus, unlike traditional Term Frequency - Inverse Document Frequency (TF-IDF) approaches which model each document with a vocabulary sized vector, topic models represent each document with a distribution over a mixture of K topics. In Table 1.1, we show a subset of learned topics on the Wikipedia 2014 corpus, along with the most probable words for each topic in vertically descending order.

Latent Dirichlet Allocation (LDA) [26] is a generative process to explain how documents are written. The name Dirichlet stems from the fact that document vector priors are drawn from the Dirichlet distribution. It makes two key assumptions. Firstly, it assumes that each document is independently generated. In other words, when we observe one document, this observation does not influence the observation of other documents in the corpus. Secondly, it assumes the exchangeability of words in a document. Exchangeability is a statistical notion stating that for a set of random variables,

any reordering of them to get a new sequence does not change the probability of the document. In this regard, it discards a certain number of grammatical dependencies between the words and enables high-level descriptive summary of documents. Both assumptions oversimplify the document generation process but have shown to perform well and capture meaningful topics.

The generative process of the LDA is illustrated as a graphical model in Figure 1.3 and formally can be described as follows:

- Sample document's topic distribution $\boldsymbol{\theta} \sim Dir(\boldsymbol{a})$
- for each word w_n in the document:
  - Sample a topic $\boldsymbol{z}_n$ from $Mul(\boldsymbol{\theta})$.
  - Sample a word from $p(w_n|\boldsymbol{z}_n, \boldsymbol{\beta})$.

where $Dir(\boldsymbol{a})$ represents a Dirichlet Distribution and $Mul(\boldsymbol{\theta})$ is the Multinomial distribution. $\boldsymbol{\alpha}$ is the governing parameter for the topic distribution of documents, and $\boldsymbol{\beta}$ is the topic-word matrix where each row is a multinomial word distribution for a particular given topic. We first draw a $\boldsymbol{\theta}$ vector from the Dirichlet topic distribution. The document representation $\boldsymbol{\theta}$ vector, governs how likely it is to exhibit a particular topic for that document. The generative model then draws a latent topic indicator $\boldsymbol{z}_n$ for every word in the document. It then conditions on the given topic-word distribution $\boldsymbol{\beta}$ to sample words $\boldsymbol{w}_n$ in the document. Here, unlike word indicators w_n that are observed random variables, $\{\boldsymbol{\theta}, \boldsymbol{z}\}$ are latent random variables. As we get more and more documents, we update the latent values $\{\boldsymbol{\theta}, \boldsymbol{z}\}$ such that these parameters explain the observed words. The full posterior of this probabilistic model combines the likelihood and the prior of the LDA:

$$p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{n=1}^{N} p(w_n|\boldsymbol{z}_n, \boldsymbol{\beta}) p(\mathbf{z}_n|\boldsymbol{\theta}) \tag{1.2}$$

where for simplicity, we assumed that each document has N words.

In its full generality, maximization of the LDA's posterior distribution in Equation 1.2 requires approximations. Similarly to the Expectation Maximization (EM) algorithm, a distinction between inference and learning is made. In the maximization step, $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ parameters are learned. In the inference step, the latent random variables $\{\boldsymbol{\theta}, \boldsymbol{z}\}$ that maximize the posterior distribution are inferred [1]. In literature, there are many ways to perform this inference[2]; such as variational approximations [27], Monte Carlo based Gibbs Sampling [28], or hybrid variants [29].

There are many extensions to the original LDA model. For example, correlated topic models [30] alleviate the assumption of independent topics. Others applied topic models to other modalities such as images or time-series data. For instance, Zhou et al. proposed a temporal topic model in which topics represent time trajectories [31]. Hospedales et al. applied topic models to cluster motion patterns and detect detect anomalies in a

[1] Obtaining latent values $\{\boldsymbol{\theta}_{te}, \boldsymbol{z}_{te}\}$ for a given test document is straightforward, it only requires a single Expectation step $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$

[2] A variational approximation for the model is provided in Appendix A.1

Table 1.1: Highest probability words of a random set of learned topics on Wikipedia 2014 data. We observe that topic 1-3 collected words in a biology and physics context respectively. Topic 7 specialized in representing educational words whereas Topic 8 learned representation for sports words.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
cell	align	space	ireland	game	published	education	league
chemical	text	earth	irish	games	book	research	football
protein	style	star	horse	player	books	students	club
cells	dnf	nuclear	dublin	video	isbn	institute	round
acid	bar	energy	stakes	players	press	science	cup
gene	colspan	physics	northern	version	author	department	player
dna	color	solar	lengths	chess	works	professor	games
structure	center	light	race	super	magazine	association	teams
chemistry	right	science	derby	character	journal	award	tournament
reaction	linear	sun	cork	characters	editor	director	game

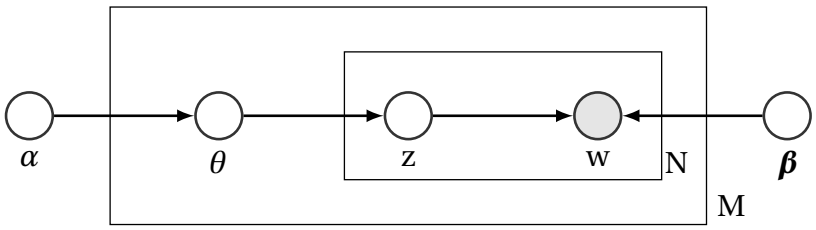


Figure 1.3: Graphical illustration of the LDA Model using the plate notation. Shaded nodes indicate observed variables. Unshaded nodes are random variables. Arrows indicate probabilistic conditioning relation. Rectangular plate notation represents repetitions. Each outer box is a copy for M documents, and the inner box is for each word in the corresponding document.

visual surveillance scenario [32]. In this case, topics represent low-level motion (optical flow), and topic distributions correspond to high-level behavior, which is temporally connected across video clips. Griffiths et al. proposed a topic model which targets to capture syntactic structure [33]. They build a joint temporal topic model where LDA component captures long term dependencies, and a Hidden Markov Model (HMM) component captures short term interactions. Blei et al. proposed a dynamical topic model to model the temporal evolution of documents with a Gaussian state transitions over time [34]. Emonet et al. proposed an extended motif model for modelling spatio-temporal word (flow) co-occurrences [35]. Mainwright et al. showed how the LDA model can be conveniently reparameterized in the exponential family model [36]. For a comprehensive literature overview on topic models, we refer the interested readers to [37].

1.2.3. MARKOV RANDOM FIELDS

Similarly to word embeddings and topic models, a Markov Random Field (MRF) based embedding is another powerful exponential family model [36, 38]. This probabilistic model consists of observed and latent random variables. The observed random variables are measurements. Since real world measurements are not usually precise, measurements are assumed to be a noisy realization of the underlying latent random variables. In Figure 1.4, we illustrate the graphical model of an MRF.

The MRF model then relates the observed variables to latent variables via potential functions. These functions measure the amount of consistency between two variables.

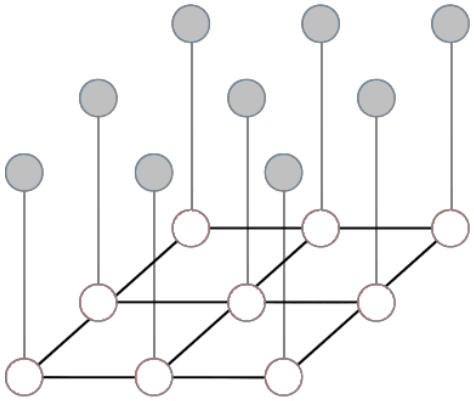


Figure 1.4: Graphical illustration of an MRF Model on a 3x3 image using the plate notation. Shaded nodes indicate observed variables. Unshaded nodes are latent random variables. Each link indicates a potential function.

The learning of the random field means performing probability maximization over states of latent random variables to find consistent states for them and the whole random field. Intuitively, we aim to find a latent representation that is likely to be the noiseless version of the observation while latent variables are consistent with their neighbors. We provide a broad explanation of the MRF model in Chapter 7.

Originally, MRFs have found their usage in image-based applications. Some examples include medical image registration [39], a variety of texture modeling applications [40], ranged sensing [41], and image denoising [42]. Recently, they have been applied to text based applications. Chen et al. showed how text segmentation can benefit from random field optimization [43]. Faruqui et al. recently applied an MRF as a postprocessing technique to improve the quality of the word embeddings [44]. They first extract relational information from semantic lexicons and construct a random field over the words of the vocabulary. After learning the word embeddings on large corpora, they treat them as noisy observations, and they then refine each learned word vector by minimizing the energy over the random field.

In Section 1.2 we elaborated that noise is an essential characteristic of the text data. Due to this, we adopted a probabilistic methodology. Although MRFs have been applied to the text problems successfully, we questioned whether random field modeling help for other domains where substantial noise is present in the data? In Chapter 7, we search an answer to this question and present our results on modelling wind energy measurement data. We show that an MRF model is able to fuse several wind farm suitability factors, each exhibiting a different amount of measurement noise, to determine which regions are more promising for establishing wind farms.

1.3. EXPONENTIAL FAMILY REPRESENTATIONS

Compared to traditional knowledge-based approaches in artificial intelligence, probabilistic modelling techniques offer a lot for learning representations of data. These probability models assume that observations can be represented with particular probability distributions. Assumptions in probabilistic modelling have to deal with two dilemmas.

Firstly, the model family has to be sufficiently large to have a rich number of model instances so that it can represent the intrinsic aspects and variations of the data at hand. Secondly, the model family should be still simple enough such that model parameters are confidently estimated.

All models, including word embeddings, topic models, and MRFs we deal with in this thesis are exponential family models. In its full generality, we consider an exponential family for modelling the data x which contains the set of probability density and mass functions in the form:

$$p(x|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} h(x) \exp(\boldsymbol{\theta}^T \phi(x)) \tag{1.3}$$

where $\boldsymbol{\theta}$ represents the *natural parameters* (or canonical parameters) which we would like to learn. $\phi(x)$ is a vector of *sufficient statistics* of the data. $h(x)$ is a scaling constant[3] and $Z(\boldsymbol{\theta})$ is a partition function of the model. The exponential family poses a unified view of different continuous and discrete distributions by using a canonical representation. It is often chosen in modern machine learning models. We list a few reasons why they are chosen:

- Given a set of constraints, the exponential family naturally arises as the solution from the set of distributions which makes the least number of assumptions for the maximum entropy problem[4].
- It is known that the exponential family has finite-sized sufficient statistics. This property imbues models of the family with the ability to summarize a large number of independent and identically distributed samples using only a small set of numbers. With their finite sized sufficient statistics, there is no loss of information of the data [37].
- Bayesian statistics advocates the use of conjugate priors in the likelihood function which greatly simplifies the calculation of the relevant posterior distributions. One exponential family naturally comes with its conjugate prior [46] and consequently makes the family very suited for learning purposes.

Moreover, many simple building block probability distributions such as the Normal, Exponential, Gamma, Bernoulli, Poisson and Dirichlet distributions, can be re-parameterized to be in a particular exponential family. For more background on the exponential family of probability distributions, we refer the interested readers to the seminal paper of [36].

1.4. FOCUS OF THIS THESIS

This thesis focuses on exponential family word embeddings in two aspects; improving the representative power of word embedding models, and developing efficient learning strategies for them.

[3] Some exponential family notations absorb the scaling function into the exponential.
[4] A simplified proof of Maximum Entropy problem is in [45].

- Availability of training data is important for training word embeddings. While text datasets for natural languages such as English Chinese and Spanish are publicly available, for most natural languages [47] there is not a lot of data available. For such scenarios, we must ensure that the embedding model generalizes well despite the small amount of data available for training.

  Research Question 1 *How can we learn more robust representations given scarce text datasets?* (Chapter 3)

- For some languages a vast amount of training data is available, but across multiple lexical resources which all have varying levels of semantic informativeness. Hence, one would like to effectively mix these available data taking their informativeness into account.

  Research Question 2 *How to fuse lexical sources with varying structure to specialize embeddings for semantics?* (Chapter 4)

- Word embedding models assume each word can be represented with one particular unique sense and thus do not address polysemy. Our hypothesis is that we can automatically distinguish different word senses from the given context. Thus, a new embedding model with a novel objective function to take polysemy into account can be derived.

  Research Question 3 *How can we learn word representations taking polysemy into account?* (Chapter 2)

- It is a very common practice to employ a negative sampling approach for learning word embeddings. However, it includes heuristic specifications of the sampling distribution and usually driven by empirical experience. This is a resource-demanding step and requires extensive experimentations for satisfactory performance. Optimizing the sampling distribution can eliminate faulty heuristic specifications.

  Research Question 4 *How to efficiently optimize negative sampling distributions to eliminate heuristic specifications?* (Chapter 5)

- Negative sampling is much faster compared to the maximum likelihood based estimators for learning word embeddings. Nevertheless, when sampling is performed with small sample sizes, an accurate estimation of the denominator in Equation 1.1 turns out to be problematic. This drawback can be addressed by providing further guidance to the sampling step during the word vector training.

  Research Question 5 *How to develop a word embedding sampler that is a more reliable estimator of the partition function?* (Chapter 6)

- MRF has been applied to the noisy text problems successfully. It is interesting to investigate whether MRFs can also deal with the measurement noise in other domains such as wind energy farm placing. In this scenario, the suitability criterion exhibit different amount of measurement noise and utilization of a random field can have positive consequences and immediate effects on wind farm decision making.

Research Question 6 *Does MRF help decision making in wind energy farm placing? (Chapter 7)*

2

PAWE: POLYSEMY AWARE WORD EMBEDDINGS

2.1. ABSTRACT

Word embedding models learn a distributed vectorial representation for words, which can be used as the basis for (deep) learning models to solve a variety of natural language processing tasks. One of the main disadvantages of current word embedding models is that they learn a single representation for each word in a metric space, as a result of which they cannot appropriately model polysemous words. In this work, we develop a new word embedding model that can accurately represent such words by automatically learning multiple representations for each word, whilst remaining computationally efficient. Without any supervision, our model learns multiple, complementary embeddings that all capture different semantic structure. We demonstrate the potential merits of our model by training it on large text corpora, and evaluating it on word similarity tasks. Our proposed embedding model is competitive with the state of the art and can easily scale to large corpora due to its computational simplicity.

2.2. INTRODUCTION

Distributed word embedding models are probabilistic language models in which each word is represented with a distinct high dimensional continuous vector. The prominent advantage of using such representations is the ease to measure vector similarity with simple techniques. Moreover, an intriguing property of these models is capturing several relationships between entities as vector offsets in space. For instance, the vector $v_{king} - v_{queen}$ yields a similar vector to $v_{man} - v_{woman}$, capturing the gender relation implicitly [48]. Computationally, it is possible to learn such embeddings using neural formulations that scales well with vast amount of data. Nowadays, word embeddings can be trained on corpora having billions of tokens with off-the-shelf hardware. Learned embeddings can be used for a diverse set of applications, such as information retrieval [49], machine translation [50] and sentiment analysis [7].

One property of natural languages is the polysemous words, i.e. words having multiple senses. In many languages, some words even have dozens of senses which greatly reinforces the lexical ambiguity. Despite the work in linguistics [51, 52] and psychology [53] domains to detect and resolve polysemy based ambiguities, computational embedding models usually ignore polysemy and represent each word with a single vector. This constraints the word representation to be invariant under the polysemy transformations. This is unnatural since one might expect that the representation of a particular polysemous word (e.g. *book*: a reading material (noun) or reserving a resource (verb)) to vary in different contexts.

We remove this limitation of distributed embedding models by having multiple complementary prototypes that explain possible senses of the words more naturally. A polysemy aware representation provides a more natural embedding of words and helps to disambiguate word meaning by decoupling meanings into different maps. The main contribution of this work is a new word embedding model that can appropriately represent polysemous words by learning multiple, complementary embeddings. The proposed Polysemy Aware Word Embeddings (PAWE) 1) ameliorates the representation polysemous words by learning multiple complementary embeddings, 2) retains favorable properties of prior models 3) can be trained online on large corpora. The performance of the learned embeddings is demonstrated in word similarity tasks. Experimental results show that our method successfully distinguishes different senses and learns embeddings that perform better compared to the state of the art embedding models on Wikipedia corpus. For reproducibility, we provide an open source prototype implementation of our embedding approach[1].

2.2.1. RELATED WORK

Our work is a combination of two streams of work, the construction of word embeddings and techniques addressing the polysemy property of the language.

Word Embedding Architectures. Various statistical language models are proposed to obtain word representations that generalize to multiple tasks [54, 55]. Preliminary works focused on stochastic models that have a large hidden layer with stochastic units.

[1] MATLAB+MEX implementation of the proposed model can be downloaded from http://homepage.tudelft.nl/8f9v2/poly_pawe.zip

One example is the Factored Restricted Boltzmann Machine framework whose factors represent input and predicted words [56]. Since such a generative architecture is extremely slow to train, recent work shifted to logbilinear (LBL) architectures which replace stochastic hidden variables with a simple hidden layer for producing the prediction. Continuous Bag of Words (CBoW), SkipGram [57, 58], Robust Gram [59] and Paragraph Vector [60] models can be considered as different logbilinear models. In our work, we utilize a large but computationally cheap hidden layer.

The computational bottleneck of logbilinear architectures is the softmax output layer, which requires a summation over the vocabulary to obtain a valid probability mass function. Since the softmax unit renders maximum likelihood to be expensive, the learning requires approximate inference techniques such as Importance Sampling [13] or Noise Contrastive Estimation (NCE) [61, 62]. Another strategy to avoid summation over the vocabulary is to construct a hierarchical decomposition tree using semantic priors [63]. Similar to [48], we also use the Negative Sampling variant of NCE to learn plausible embeddings.

Aforementioned logbilinear architectures can be viewed as techniques to factorize a non-negative matrix of corpus statistics into context and target matrices [19] and are related to Non Negative Matrix Factorization techniques [64]. Rather than training such architectures, some methods first extract useful statistics of the corpus (such as word co-occurrences) and discover embeddings using PCA or HPCA [65]. However such approaches suffer from the disproportionate effects of stop words such as 'the','a' in the corpus which co-occur with many words in the language. They are also very susceptible to data sparsity. In our formulation, in order to account for such effects, we apply a simple subsampling technique during the learning so that very frequent words will have a lower probability to be sampled.

Polysemy Modeling. The problem of modeling polysemy has been addressed in several different works [66, 67]. Neelakantan et al. proposes a nonparametric way to capture different meanings [68]. Tia et al. proposed a mixture model for learning multi-prototype embeddings [69]. They train a multi-prototype Skip-Gram model and train it using an EM algorithm. Since the exact solution to the maximization step is not available in their model, they use gradient descent to optimize the maximization step. Reisinger et al. employ an initial clustering step to extract different senses of words in the vocabulary [70]. Then for each word sense, a representation is learned individually. Similar multi-prototype word vector ideas are also employed in the context of neural word embeddings [71], [72], [73]. In our work, we do not perform a pre-clustering step to extract multiple meanings. We directly represent multiple embeddings using a unified logbilinear energy where various meanings are automatically discovered during the optimization. Hence, our technique avoids adjusting extra clustering parameters. Moreover, by avoiding a distinct offline clustering step, our model readily extends to new unknown senses when new senses of words are introduced during the training.

Some other works exploit additional supervised information. This is done by incorporating annotated knowledge of the senses of words from a knowledge base such as WordNet. In Chen's work [74], sense vectors are also learned along with word vectors for Word Sense Disambiguation [75] task. They show that word sense representation and word sense disambiguation tasks can benefit from each other. While it is possible to

increase the quality of embeddings with increased supervision [55], annotated sensual knowledge might not be available in a general setting (for example, such a database is missing for the Turkish language). Contrary to their work, our model does not exploit such priory knowledge. As a result, compared to other embedding approaches that has the same sample size, our model has potential to discover polysemy relations in unsupervised fashion: learning of the model does not require any extra supervision such as ground-truth polysemy or sense annotations.

From this perspective, our work is related to graphical models such as Similarity Component Analysis [76] or Latent Dirichlet Allocation (LDA) [77]. Both LDA and our model are unsupervised. LDA discovers hidden topics on document level while our method discovers different senses on word level to learn embeddings. While LDA's generative process ignores word order, it is easy to extend our formulation to account for word ordering with a simple weighting.

A closely related method to our work is the Multiple Maps T-SNE algorithm [78]. Since high dimensional non-metric pairwise similarities can not be preserved in low dimensional spaces, the authors propose multiple maps to represent intransitive non-metric similarities. Their technique conditions on given high dimensional pairwise word distances and finds low dimensional embeddings while we directly learn the high dimensional embeddings from the corpus.

2.3. DISTRIBUTED WORD EMBEDDINGS

In this section, we start by formulating distributed word embeddings. Then we describe our proposed approach, followed by its learning technique.

2.3.1. PROBLEM FORMULATION

We are given a set of vocabulary indices of words as the training dataset $D = \{d_1, d_2, .., d_x, .., d_N\}$ with N words in the corpus, d_x representing the vocabulary index of x'th word in the text. Let q denote the iterator over the vocabulary of size V and $\mathbf{w}_q$ be word q's one-hot encoded representation such that $w_{qj} \in \{0,1\}$ and $\sum_{j=1}^{V} w_{qj} = 1$. We use $|\mathbf{w}_q|$ to indicate number of times the q'th word occurs in the corpus. Let Φ, Ψ be the $D \times V$ target and context embedding matrices that map each word into a continuous D dimensional space. We would like to learn parameters $\theta = \{\Phi, \Psi\}$. In light of the distributional hypothesis (words that occur in same context tend to purport similar meanings), embedding formulations disregard long range dependencies in the text and represent the context of a word by a (small) set of surrounding words. While other definitions of context are possible, we use bilateral words for the context representation.

Let $S_x = \{\mathbf{w}_{d_{x-t}}, .., \mathbf{w}_{d_{x-1}}, \mathbf{w}_{d_x}, \mathbf{w}_{d_{x+1}}, ..., \mathbf{w}_{d_{x+t}}\} = \{\mathbf{C}_{d_x}, \mathbf{w}_{d_x}\}$ represent x'th sentence of the training set with word d_x to be predicted. The goal is to minimize the negative conditional log likelihood of the training data for all sentences:

$$\theta^* = \arg\min_{\hat{\theta}} \sum_{\forall x} \log P(\mathbf{w}_{d_x} | \mathbf{C}_{d_x}; \hat{\theta}) \tag{2.1}$$

where $\mathbf{C}_{d_x}$ is the context and defined as $\mathbf{C}_{d_x} = \{\mathbf{w}_{d_{x+i}},\ i \in \{t-1, .., -1, 1, .., t+1\}\}$ and t is the window size parameter.

In the online setting, one epoch consists of performing a single pass over the training set and performing gradient updates with an iterative optimization algorithm. For simplicity, let us focus on sentence S_x, with target word's identity $q = d_x$. The conditional probability $P(\mathbf{w}_q|\mathbf{C}_q)$ is given as:

$$P(\mathbf{w}_q|\mathbf{C}_q) = \frac{\exp(s_\theta(\mathbf{w}_q, \mathbf{C}_q))}{\sum_{\tilde{q}} \exp(s_\theta(\mathbf{w}_{\tilde{q}}, \mathbf{C}_q))}, \tag{2.2}$$

where $s_\theta(\mathbf{w}_q, \mathbf{C}_q)$ is called the score function in statistics. Different score functions yield different logbilinear models which will be discussed in the next section. First we describe our embedding model.

2.3.2. PAWE EMBEDDING MODEL

Word embedding techniques learn different embeddings based on their predictive formulation. The CBoW, Skip Gram [57], GloVe [79] and Paragraph Vector [60] models all have different $s_\theta(.)$ functions. While we base our model on CBoW architecture due to its simplicity and speed, we note that it is equally applicable to other embedding methods, thanks to its generic formulation. The single prototype CBoW score function is given as:

$$s_\theta(\mathbf{w}_q, \mathbf{C}_q) = \frac{1}{2t}\left(\sum_{\mathbf{w}_r \in \mathbf{C}_q} \left(\Phi\mathbf{w}_r\right)^T (\Psi\mathbf{w}_q)\right) + b_q, \tag{2.3}$$

where $\Phi\mathbf{w}_r, \Psi\mathbf{w}_q \in \mathscr{R}^D$ are the context and target embeddings obtained by projecting the one-hot encoded representations onto the embedding spaces, and where b_q is the prediction bias of $\mathbf{w}_q$. For the sake of simplicity, we will drop the b_q and $\frac{1}{2t}$ from the notation.

Because the score of the CBoW model (Eq. 2.3) penalizes the dissimilarity between $\mathbf{w}_q$ and the arithmetic mean of the context word embeddings around this word $\mathbf{w}_q$, words that appear in the same context, should be close in high dimensional space as well. However, knowing that each word can have multiple senses, the same prediction will be used to penalize possibly different senses of a target word $\mathbf{w}_q$.

A better score function must take polysemous cases into account and automatically compute a score for multiple senses of a target word. This can be done by creating multiple prototypes of a target word and representing the target-context similarity as a weighted sum. Following this idea, we propose a score function that takes polysemous cases into account whilst staying computationally efficient. Formally, the score of the Polysemy Aware Word Embeddings embedding model is defined as:

$$s_\theta(\mathbf{w}_q, \mathbf{C}_q) = \log \sum_m \pi_q^m \exp\left(\sum_{\mathbf{w}_r \in \mathbf{C}_q} (\Phi\mathbf{w}_r)^T (\Psi^m \mathbf{w}_q)\right), \tag{2.4}$$

where m is the index to iterate over the M prototypes and π_q^m is the weight of $\mathbf{w}_q$'s m'th prototype. Each word weight denotes how important a particular sense is in an individual map. These weights can also be interpreted as prior probabilities of occurence of the different word senses in a corpus. The score function finally combines the prediction scores for each map using a weighted linear combination.

Ideally we would like each prototype weight to be bounded on an interval, but introducing constraints per word to the formulation complicates and slows down the optimization. Instead we optimize unconstrained weights $\mathbf{W}_q^m$ and make π depend on the unconstrained weights using the sigmoid function: $\pi_q^m = \sigma(\mathbf{W}_q^m)$. We have experimented with few other functions to constrain the map weights and found that the sigmoid function works best.

Conceptually, Equation 2.4 can be interpreted as a mixture model, with an unnormalized prior distribution. In this sense, if the target word has several distinct senses (e.g. jaguar), one representation will quickly specialize to represent one particular meaning with updates to its weight. The model will still be able to represent cases in which each word has only one meaning. Despite the fact that the number of maps M has to be specified beforehand, in practice the model does not behave like a hard clustering method that forces each word to have a predefined number of meanings. For PAWE model, the parameter vector consists of $\theta = \{\Phi, \Psi_{1:M}, \mathbf{W}\}$ where M is the number of target maps used and $\mathbf{W}$ is $M \times V$ unconstrained word weight matrix.

The PAWE model is a more general case of the LBL model, and it boils down to the single prototype model when the number of maps is equal to one. Applying multiple maps to both Φ and Ψ introduces a high degree of parameter redundancy and makes learning relatively harder. Armed with this knowledge, we only represent the target embedding Ψ with multiple maps. Doing so also prevents overfitting in the training of our model.

2.3.3. LEARNING

Since our model is from the family of probabilistic models, it shares the same bottleneck: during the optimization, evaluating partition function of the distribution requires summing over the whole vocabulary (Eq. 2.2), which quickly becomes problematic for large vocabularies. This yields Maximum Likelihood Estimation approach very expensive to use. Even for a single word update $\mathbf{w}_q$, the gradient $\partial J_q(\theta)$ requires a full pass over the vocabulary set, with a training complexity of $O(S_x \times V)$. Indeed, it is possible to approximate this update with algorithmic approximations such as Hierarchial Softmax. However, this approximation technique requires construction/learning of a tree on the vocabulary which is yet another difficult learning problem to address.

We bypass such difficulties by resorting to a new estimator: called Negative Sampling approximation [48]. The key idea of negative sampling learning is to train a logistic regressor to distinguish samples arising from data and samples from the noise distribution. Negative Sampling estimation is an instance of *Unsupervised as Supervised Learning* algorithms [80]. For word embeddings, we obtain noise samples by randomly changing words of sentences. For one training sample $\{\mathbf{w}_q, \mathbf{C}_q\}$, the contribution to the total cost $J(\theta)$ is:

$$J_q(\theta) = \mathbb{E}_{P_d}[\log\Big(\sigma(s_\theta(\mathbf{w}_q, \mathbf{C}_q))\Big)] + \mathbb{E}_{P_n}[\log\Big(\sigma(-s_\theta(\mathbf{w}_n, \mathbf{C}_q))\Big)] \tag{2.5}$$

where the second term is the expectation over the noise distribution P_n. Practically, the expectation is approximated by sampling a few negative instances from the noise distribution. The noise distribution is usually chosen to be a distribution over unigrams, that is proportional to the occurrence frequencies of the unigrams raised to some power: For

English embeddings trained with a highly scientific content language such as Wikipedia, $P_n(\mathbf{w}_n) \propto |\mathbf{w}_n|^{0.75}$ is known to work best which we also validated by tuning the exponent parameter in the range $[0.5, 1]$. An empirical justification to the raised power is provided in [79].

The gradient of Eq. 2.5 with respect to θ is given by:

$$\begin{aligned}\frac{\partial}{\partial\theta} J_q(\theta) = \Big(1 - \sigma(s_\theta(\mathbf{w}_q, \mathbf{C}_q))\Big) \frac{\partial}{\partial\theta} s_\theta(\mathbf{w}_q, \mathbf{C}_q) \\ - \sum_{k=1}^{K} \left[\sigma(s_\theta(\mathbf{w}_k, \mathbf{C}_q)) \frac{\partial}{\partial\theta} s_\theta(\mathbf{w}_k, \mathbf{C}_q) \right]\end{aligned} \tag{2.6}$$

where K is the number of negative samples used in practice. For our PAWE model, the parameter vector consists of $\theta = \{\Phi, \Psi_{1:M}, \mathbf{W}\}$ where M is the number of target maps used and $\mathbf{W}$ is $V \times M$ unconstrained word weight matrix. Since it is very difficult to tune the learning rates of Stochastic Gradient Descent, we instead learn the parameters with Adagrad. The idea is simply to store the historical gradients from previous steps of the optimization, and use these to automatically tune the learning rate:

$$\theta_i(t+1) = \theta_i(t) - \eta \frac{g_i(t)}{\tau_0 + \sqrt{H_i(t)}} \tag{2.7}$$

where $\theta_i(t)$ is the i'th parameter value at t'th step of the optimization and $g_i(t)$ is its gradient. η is the master step size that is less sensitive compared to the Stochastic Gradient Descent learning rate. $H_i(t)$ is the historical gradient that is $H_i(t) = \sum_{r=1}^{t} g_i(r)^2$. $H_i(t)$ is then recursively updated at every step of the optimization as follows:

$$H_i(t) = H_i(t-1) + g_i(t)^2 \tag{2.8}$$

Since Adagrad's learning rate is adapted component-wise, optimization adapts to the curvature of the loss function more precisely. The historical component of the denominator adjusts whether more updates are required to reach the minimum. For our model, with negative sampling approximation, the computational complexity is reduced to $O(S_x \times k \times M)$ and scales linearly with the number of maps.

2.4. EXPERIMENTAL RESULTS

Setup and Training Protocols. We trained PAWE on the Wikipedia 2006 and Wikipedia 2014 corpora having 100M and 3B tokens respectively. For each year multiple snapshots are provided, we selected snapshot-20141208. We use standard preprocessing protocols: the HTML tags and non visible text are removed, content is lowercased, reducing it to word tokens. We compare our model with the CBoW baseline models. For all models, we use AdaGrad for optimization with a master step size of 0.05. The minibatch size for all experiments is set to 1. The window size parameter t is set to 4. Rest of the parameters follows the standards in [57]. We set the number of negative samples to number of maps for each experiment. Unlike in the GLoVe model, we do not perform any post training operations on embeddings (such as $\Psi + \Phi$) and simply use $\Psi_{1:M}$ as the output embeddings.

Table 2.1: Nearest neighbors of some polysemous words are shown for each map.

	Map 1	**Map 2**
memory	processor,processors,mode	pupil,gift,pleasure
elementary	graduate,school,schools	theorems,geometry,thermodynamics
show	contain,survive,appear	club,host,bbc
shows	commercials,selling,mtv	represents, gives, presents
site	website,com,forum	monument,tallest,canal
bill	floyd,charlie,tom	jury,lawsuit,court
resolution	amendment,statute,amendments	frequency,bandwidth,output
press	fbi,editorial,scandal	routledge,ed,journal

We have experimented with a grid of values to determine the number of target maps, M. We first queried the English lexical database of WordNet [81], which contained a vocabulary of 147k words with a total number of 316k total senses. This reported an average of 2.15 polysemy amount per word. For our experiments with $M > 2$ the overall results slightly increased for our architecture, which is consistent with the average sense statistics of the English corpora. As the number of senses a word has follows the power law, marginal benefit from our model decreases with increased number of maps.

2.4.1. NEAREST NEIGHBORS ANALYSIS.

For this task, we randomly sampled words with replacement from the vocabulary and selectively rejected words that we believe with a high confidence does not contain multiple senses. This resulted in a subset the words that we can inspect its nearest neighbours in space. We measure the cosine distance [19] to show polysemous words' neighbours.

The discovered polysemy relations are demonstrated in Table 1. The bolded words in the first column are the query words and each column in the table depicts three nearest neighbor of a query word in a particular map. The obtained neighbours in many rows indicate that the model is capturing different senses of a word. The interesting observation here is that it sometimes pools semantic and syntactic regularities of a word into different maps. For example, for the word *resolution*, the first map captures the meaning used in a legal context, and the second map captures the meaning in the technological sense. For the word *shows*, second map captures the syntactic tense relation where first map only discovered the medial sense that is a semantic relations. We also analyzed the rejected words' nearest neighbours but do not report them since they were not pretty much informative, and mostly identical in all maps.

It must be noted that we do not constrain the model with explicit supervision to discover these regularities. When these regularities are inherent in the data, our model automatically discovers them. As there is no supervision of these maps, they do not necessarily capture a particular semantic or syntactic context such as document topics. In topic modeling approaches where each topic exhibits a particular meaning aspect, this is not the case in our polysemy aware model. Rather, lingual regularities occurs weakly on the word level.

It is difficult to visualize how our high dimensional vectors are distributed in the space. We project our vectors using t-SNE data visualization in order to analyze how

multiple maps look like. Since it is impossible to inspect all the vocabulary, we selected few polysemous words: *paper* and *size* that is extensively studied in the work of [51]. We then inspect the local neighbourhood of these words. In Figure 2.1, we observe that for the word *paper*, the first map captures material sense of *paper*. The nearest neighbours of this sense are the substances or fabrics used in the production of clothes, furnitures and buildings. The latter map captures the academic sense of the paper with neighbours such as researches, discussion, document, approach. Notice that there is almost none common neighbours in both maps, suggesting that these points capturing the senses of the *paper* are far away in the high dimensional space as well.

In Figure 2.2, we depict the case for word *size* where first map captures the physical sense. This sense represents the greatness of a physical quantity such as the rotation amount, trajectory length, or plane width. The second map captures the concept of geographic *size* that is a measure of the temperature and natural disasters. We observe no common neighbours also for this word.

2.4.2. WORD SIMILARITY

The second evaluation of the embeddings is to check the correspondence between human similarity judgements of words, and the cosine similarity in the embedded spaces. We use two datasets: the WordSim-353 dataset [82] and the Stanford Contextual Word Similarities (SCWS) dataset [71]. The WordSim-353 dataset consists of 353 pairs of nouns. For each pair, a relatedness measure is assigned by 13 to 16 human judges, 0 indicating that no relation is present and 10 indicating the maximum similarity. In order to measure the correlation between embedding and human similarity judgements, we use Spearman's Correlation Coefficient.

WordSim-353 dataset is a standard evaluation set for word embeddings and do not necessarily contain polysemous word pairs. In contrast, SCWS dataset contains 2003 pairs of words and designed to reflect interesting variations of homonymous and polysemous words. For each word pair, Part-of-Speech (POS) tags and a long sentence is provided to disambiguate the meaning of each word. Ten individual human ratings judge the similarity of the word pairs.

We define the similarity of multiple prototype vectors using the AvgSimC metric as in [71]:

$$AvgSimC(w_1, w_2) = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} P(\mathbf{w}_1^i|\mathbf{C}_1) P(\mathbf{w}_2^j|\mathbf{C}_2) d(w_1^i, w_2^j) \tag{2.9}$$

where $P(\mathbf{w}_1^i|\mathbf{C}_1)$ is the likelihood of using $i'th$ prototype of word $\mathbf{w}_1$ given the sentential context $\mathbf{C}_1$, and $d(\mathbf{w}_1^i, \mathbf{w}_2^j)$ is the distance metric chosen as the cosine similarity in the embedded space. AvgSimC gives higher score when two words have similar prototypes.

Quantitative Results. Our single map baseline model is denoted as LBL (CBoW) [58]). We first ask, which words in WordSim353 are problematic for baseline LBL model. Figure 2.3 shows the results for the baseline model for a few word pairs, compared to the results of PAWE. In vertical axis we depict the normalized error, i.e. the difference between the human similarity judgements and the cosine similarity predicted by the models. For Max score, we compute similarity of multiple map embeddings and select the map having max score. Indeed, we observe that the single map model has the highest

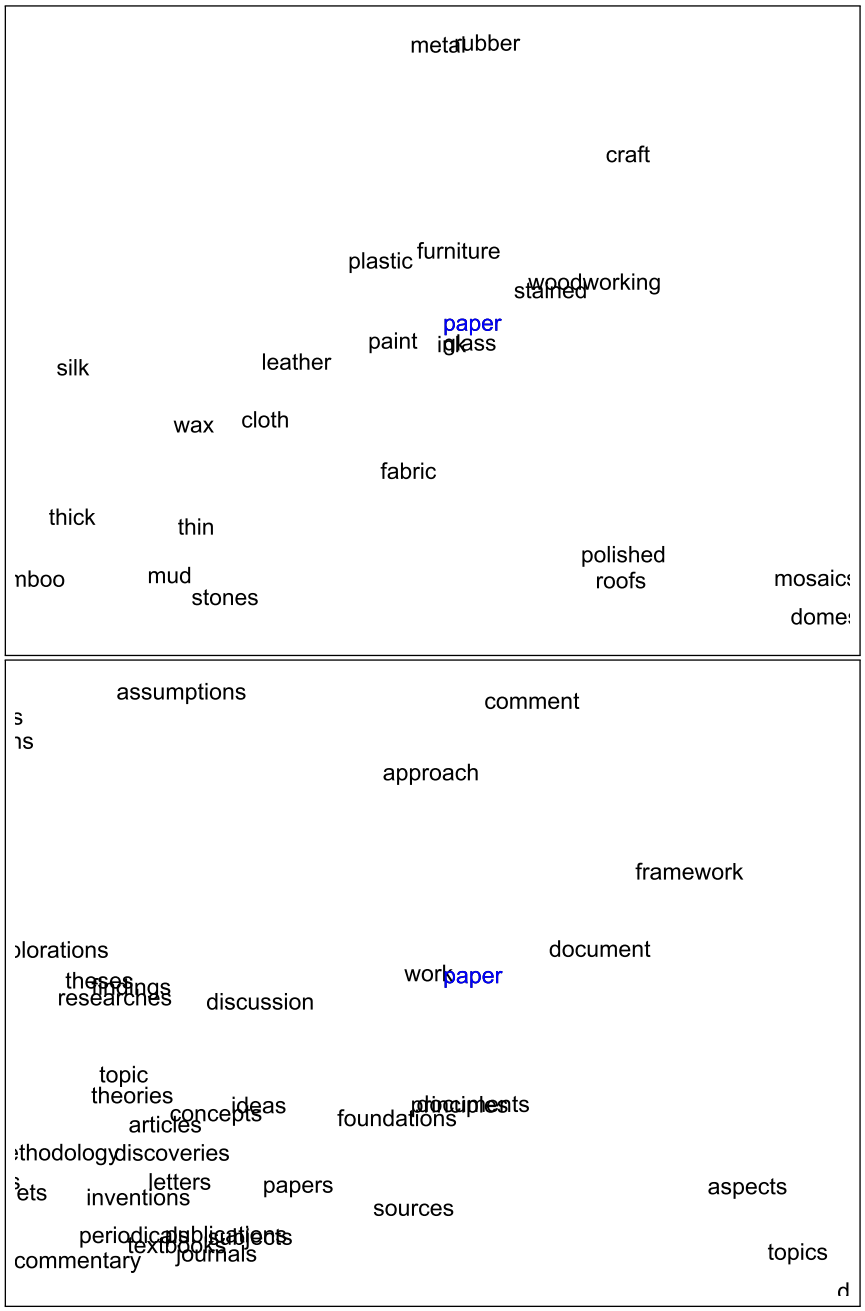


Figure 2.1: T-SNE visualization of our vectors.

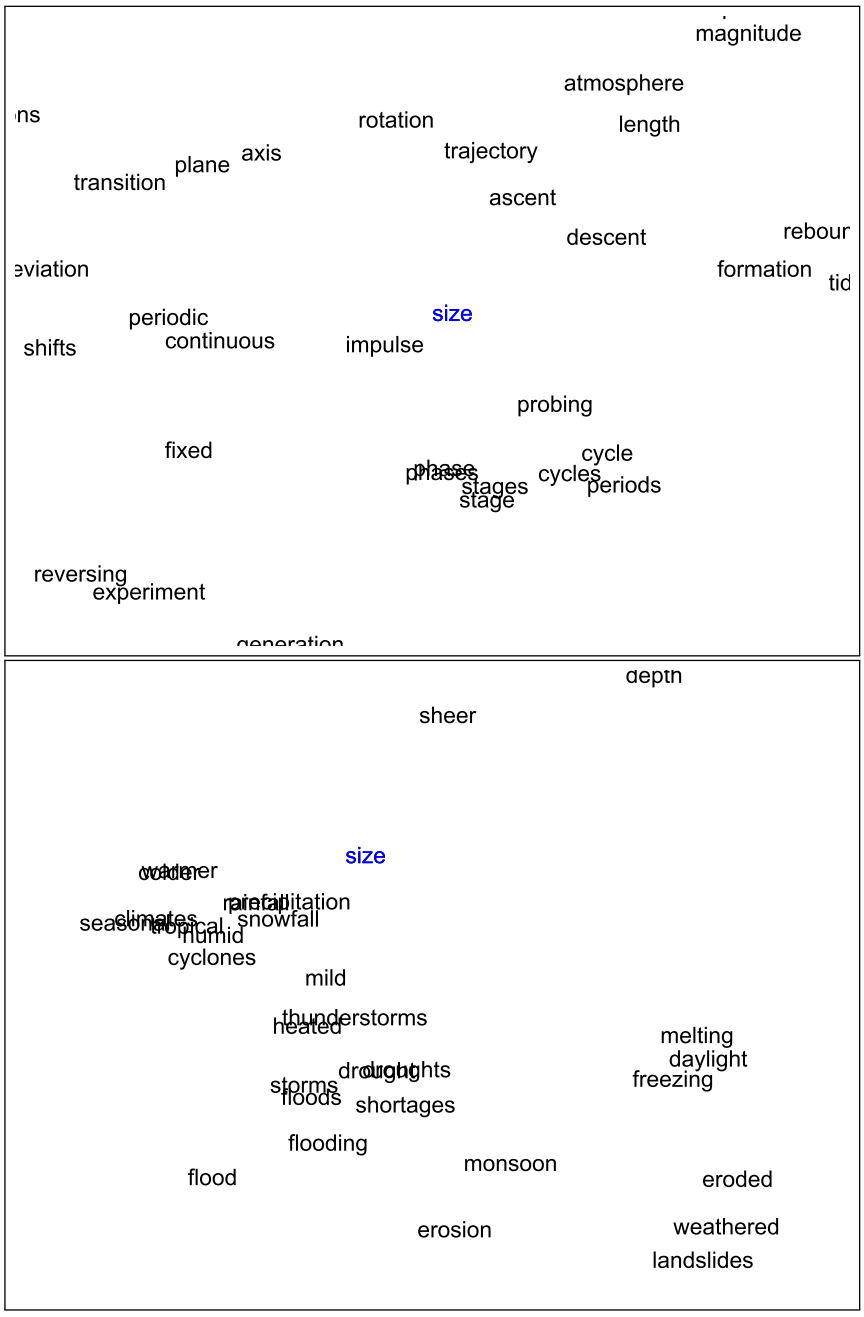


Figure 2.2: T-SNE visualization of our vectors.

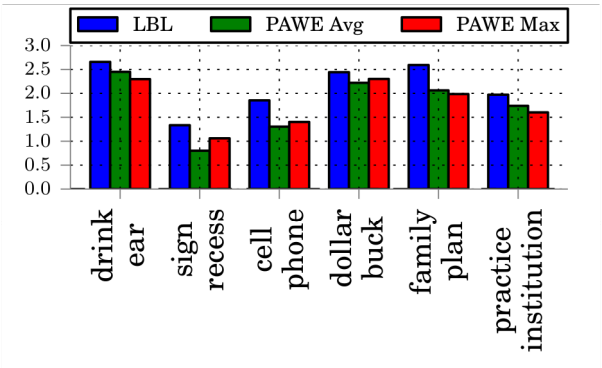


Figure 2.3: WordSim353 words having highest error.

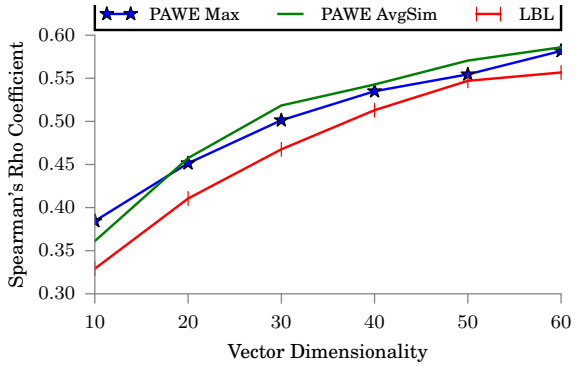


Figure 2.4: Spearman Correlations (SC) for WordSim353.

error with respect to similarity scores for word pairs including polysemous words such as "cell","family", "practice" and "sign". For all such polysemous words, our approach yields lower error compared to the LBL baseline.

To investigate the influence of the map embedding dimensionality D, we trained several models on the WordSim353 dataset and plotted the correlation as a function of the embedding dimensionality. The results are depicted in Figure 2.4. A higher dimensionality tends to increase the correlation, although the computational effort starts to increase as well. Inspecting the similarity predictions for word pairs reveals that for pairs with multiple senses (e.g. "jaguar-cat" and "jaguar-car"), the multiple map model correlates better to the human based similarities. However, while adopting our model still leverages overall similarity prediction accuracy, the margin between the baseline and our model is not quantitavely very large in this dataset. This result is expected since the WordSim353 evaluation set is a standard set for word similarity tasks and the fraction of polysemous words in query pairs is quite limited. We also evaluated the quality of the models by computing Negative Log Likelihood (NLL) on a validation subsets of Wikipedia 2014 using a 10 fold cross validation averaging. We have measured that the NLL is %2.2 lower for our embeddings, which is an indicator of an improvement over the single prototype model: our embeddings can yield a better minimum for the training objective and is more flexible to variations in the unseen data.

The word sense representation power of our model becomes more distinctive on the SCWS dataset. PAWE obtains 63.2% accuracy using AvgSimC, beating the single proto-

Table 2.2: Runtime and SC for WordSim353 and SCWS.

Model	Runtime (h)	WS353	SCWS
CB [55]	400+	55.3	57.0
LBL [57]	**0.2**	55.7	58.9
Huang [71]	168	57.9	63.5
PAWE MaxSim	**0.2** $\times M$	58.1	62.2
PAWE AvgSimC	**0.2** $\times M$	58.5	63.9

type model having 58.3% accuracy by a large margin (Table 2.2). Adapting to the polysemy, it softly votes up the suitable prototype in a context. LBL and CB baselines can not yield high performance on SCWS due to no explicit polysemy modeling. While Huang et al. achieve higher accuracy than our MaxSim instance, even MaxSim instance of PAWE is significantly faster (150x than Huang's on $M = 5$) compared to the competitor methods. Since training these embeddings on very large corpora already takes days without any special GPU hardware, our multi-prototype embeddings trades off a little accuracy but retains linear time training complexity and big data applications. The selection of number of maps is not as difficult as in a clustering setting or number of topic selection. This is because there exists strong prior knowledge in each language to select the M parameter. Also, unlike non-parametric word embedding approaches that is known to be very difficult to train [68], our model is trained similarly to LBL instances and training increases with linear time, with no extra hyperparameters guiding the training. These results suggest that composition of multiple prototypes is an appropriate representation for a variety of words in the vocabulary, suggesting that PAWE instances are a good promise between speed and accuracy.

2.5. CONCLUSIONS

This chapter presents a novel word embedding model that uses multiple complementary maps to represent the words. We developed a new score function to enable the representation of polysemous words, i.e. words that can have multiple meanings. Because polysemous words are inherent in natural languages, it is crucial that word embeddings allow for these distinct meanings. During the learning of this Polysemy Aware Word Embeddings (PAWE) we automatically discovered multiple meanings without using any human annotations for polysemy. This approach of polysemy modeling collected syntactic and semantic variations inherent in the natural language into different vector maps.

Unlike previous approaches, ours does not use any supervised entity-relationship knowledge to learn word senses, and does not utilize any clustering step. This removes the burden of tuning extra model-specific parameters for word embeddings. The algorithmic complexity of PAWE increases only linearly with the number of maps, and is comparable to the complexity of the baseline logbilinear model. Due to these reasons, the proposed embedding model is easily scalable to large text corpora containing rich polysemy collections.

We validated the behaviour and potential of our model with experimental results, and verified that the similarity based language tasks enjoy the potential of multi-prototype embeddings. For word similarity task, competitive performances are obtained using our vectors. While we demonstrated the value of such embeddings on discovering language polysemy, additional merits of learning multiple map representations is open question and left for future investigation. Prosperous directions of our future work includes exploring the theoretical gains of using multiple prototypes word vectors, effectively determining the number of prototypes for a language, and application of similar score functions to Recursive Neural Network (RNN) based language models.

3

ROBUST GRAM EMBEDDINGS

3.1. ABSTRACT

Word embedding models learn vectorial word representations that can be used in a variety of NLP applications. When training data is scarce, these models risk losing their generalization abilities due to the complexity of the models and the overfitting to finite data. We propose a regularized embedding formulation, called *Robust Gram* (RG), which penalizes overfitting by suppressing the disparity between target and context embeddings. Our experimental analysis shows that the RG model trained on small datasets generalizes better compared to alternatives, is more robust to variations in the training set, and correlates well to human similarities in a set of word similarity tasks.

3.2. INTRODUCTION

Word embeddings represent each word as a unique vector in a linear vector space, encoding particular semantic and syntactic structure of the natural language [83]. In various lingual tasks, these sequence prediction models have shown superior results over the traditional count-based models [84]. Tasks such as sentiment analysis [7] and sarcasm detection [85] enjoys the merits of these features.

These word embeddings optimize features and predictors simultaneously, which can be interpreted as a factorization of the word cooccurence matrix C. In most realistic scenarios these models have to be learned from a small training set. Furthermore, word distributions are often skewed, and optimizing the reconstruction of $\hat{C}$ puts too much emphasis on the high frequency pairs [19]. On the other hand, by having an unlucky and scarce data sample, the estimated $\hat{C}$ rapidly deviates from the underlying true cooccurence, in particular for low-frequency pairs [86]. Finally, noise (caused by stemming, removal of high frequency pairs, typographical errors, etc.) can increase the estimation error heavily [87].

It is challenging to derive a computationally tractable algorithm that solves all these problems. Spectral factorization approaches usually employ Laplace smoothing or a type of SVD weighting to alleviate the effect of the noise [88]. Alternatively, iteratively optimized embeddings such as Skip Gram (SG) model [57] developed various mechanisms such as undersampling of highly frequent hub words apriori, and throwing rare words out of the training.

Here we propose a fast, effective and generalizable embedding approach, called Robust Gram, that penalizes complexity arising from the factorized embedding spaces. This design alleviates the need from tuning the aforementioned pseudo-priors and the preprocessing procedures. Experimental results show that our regularized model 1) generalizes better given a small set of samples while other methods yield insufficient generalization 2) is more robust to arbitrary perturbations in the sample set and alternations in the preprocessing specifications 3) achieves much better performance on word similarity task, especially when similarity pairs contains unique and hardly observed words in the vocabulary.

3.3. ROBUST GRAM EMBEDDINGS

Let $|y| = V$ the vocabulary size and N be the total number of training samples. Denote x, y to be $V \times 1$ discrete word indicators for the context and target: corresponding to the context and word indicators c, w in word embedding literature. Define $\Psi_{d\times V}$ and $\Phi_{d\times V}$ as word and context embedding matrices. The projection on the matrix column space, Φx, gives us the embedding $\vec{x} \in \mathscr{R}^d$. We use Φx and Φ_x interchangeably. Using a very general formulation for the regularized optimization of a (embedding) model, the following objective is minimized:

$$J = \sum_{i}^{N} \mathscr{L}(\Psi, \Phi, x_i, y_i) + g(\Psi, \Phi) \tag{3.1}$$

where $\mathscr{L}(\Psi, \Phi, x_i, y_i)$ is the loss incurred by embedding example target y_i using context x_i and embedding parameters Ψ, Φ, and where $g(\Psi, \Phi)$ is a regularization of the em-

bedding parameters. Different embedding methods differ in the form of specified loss function and regularization. For instance, the Skip Gram likelihood aims to maximize the following conditional:

$$\begin{aligned} \mathscr{L}(\Psi,\Phi,x,y) &= -\log p(y|x,\Phi,\Psi) \\ &= -\log \frac{\exp(\Psi_y^T \Phi_x)}{\sum_{y'} \exp(\Psi_{y'}^T \Phi_x)} \end{aligned} \tag{3.2}$$

This can be interpreted as a generalization of Multinomial Logistic Regression (MLR). Rewriting $(\Psi y)^T(\Phi x) = (y^T \Psi^T \Phi x) = y^T W x = W_y x$ shows that the combination of Φ and Ψ become the weights in the MLR. In the regression the input x is transformed to directly predict y. The Skip Gram model, however, transforms both the context x and the target y, and can therefore be seen as a generalization of the MLR.

It is also possible to penalize the quadratic loss between embeddings [89]:

$$\mathscr{L}(.) = -\log \frac{\exp(-||\Psi_y - \Phi_x||^2)}{\sum_{y'} \exp(-||\Psi_{y'} - \Phi_x||^2)} \tag{3.3}$$

Since these formulations predefine a particular embedding dimensionality d, they impose a low rank constraint on the factorization $W = \Psi^T \Phi$. This means that $g(\Psi,\Phi)$ contains $\lambda rank(\Phi^T \Psi)$ with a sufficiently large λ. The optimization with an explicit rank constraint is NP hard. Instead, approximate rank constraints are utilized with a Trace Norm [90] or Max Norm [91]. However, adding such constraints usually requires semidefinite programs which quickly becomes computationally prohibitive even with a moderate vocabulary size.

Do these formulations penalize the complexity? Embeddings under quadratic loss are already regularized and avoids trivial solutions thanks to the second term. They also incorporate a bit weighted data-dependent ℓ_2 norm. Nevertheless, choosing a log-sigmoid loss for Equation 3.1 brings us to the Skip Gram model and in that case, ℓ_p regularization is not stated. Without such regularization, unbounded optimization of $2Vd$ parameters has potential to converge to solutions that does not generalize well.

To avoid this overfitting, in our formulation we choose g_1 as follows:

$$g_1 = \sum_{v}^{V} \lambda_1 \left(||\Psi_v||_2^2 + ||\Phi_v||_2^2 \right) \tag{3.4}$$

where Ψ_v is the row vector of words.

Moreover, an appropriate regularization can also penalize the deviance between low rank matrices Ψ and Φ. Although there are words in the language that may have different context and target representations, such as *the* [1], it is natural to expect that a large proportion of the words have a shared representation in their context and target mappings. To this end, we introduce the following regularization:

$$g_2 = \lambda_2 ||\Psi - \Phi||_F^2 \tag{3.5}$$

[1] Consider prediction of *Suleiman* from *the*, and *the* from *oasis*. We expect *the* to have different vectorial representations.

where F is the Frobenius matrix norm. This assumption reduces learning complexity significantly while a good representation is still retained, optimization under this constraint for large vocabularies is going to be much easier because we limit the degrees of freedom.

The Robust Gram objective therefore becomes:

$$LL + \lambda_1 \sum_{v}^{V} \left(||\Psi_v||_2^2 + ||\Phi_v||_2^2 \right) + \lambda_2 ||\Psi - \Phi||_F^2 \tag{3.6}$$

where $LL = \sum_i^N \mathscr{L}(p(y_i|x_i, \Psi, \Phi))$ is the data log-likelihood, $p(y_i|x_i, \Psi, \Phi)$ is the loglinear prediction model, and $\mathscr{L}$ the cross entropy loss. Since we are in the pursuit of preserving/restoring low masses in $\hat{C}$, norms such as ℓ_2 allow each element to still possess a small probability mass and encourage smoothness in the factorized $\Psi^T\Phi$ matrix. As $\mathscr{L}$ is picked as the cross entropy, Robust Gram can be interpreted as a more principled and robust counterpart of Skip Gram objective.

One may ask what particular factorization Equation 3.6 induces. The objective searches for Ψ, Φ matrices that have similar eigenvectors in the vector space. A spectral PCA embedding obtains an asymmetric decomposition $W = U\Sigma V^T$ with $\Psi = U$ and $\Phi = \Sigma V$, albeit a convincing reason for embedding matrices to be orthonormal lacks. In the Skip Gram model, this decomposition is more symmetric since neither Ψ nor Φ are orthonormal and diagonal weights are distributed across the factorized embeddings. A symmetric factorization would be: $\Psi = U\Sigma^{0.5}$,$\Phi = \Sigma^{0.5}V^T$ as in [19]. The objective in Eq. 3.6 converges to a more symmetric decomposition since $||\Psi - \Phi||$ is penalized. Still some eigenvectors across context and target maps are allowed to differ if they pay the cost. In this sense our work is related to power SVD approaches [92] in which one searches an a to minimize $||W - U\Sigma^a\Sigma^{1-a}V^T||$. In our formulation, if we enforce a solution by applying a strong constraint on $||\Psi - \Phi||_F^2$, then our objective will gradually converge to a symmetric powered decomposition such that $U \approx V$.

3.4. EXPERIMENTS

The experiments are performed on a subset of the Wikipedia corpus containing approximately 15M words. For a systematic comparison, we use the same symmetric window size adopted in [93], 10. Stochastic gradient learning rate is set to 0.05. Embedding dimensionality is set to 100 for model selection and sensitivity analysis. Unless otherwise is stated, we discard the most frequent 20 hub words to yield a final vocabulary of 26k words. To understand the relative merit of our approach [2] , Skip Gram model is picked as the baseline. To retain the learning speed, and avoid inctractability of maximum likelihood learning, we learn our embeddings with Noise Contrastive Estimation using a negative sample [94].

3.4.1. MODEL SELECTION

For model selection, we are going to illustrate the log likelihood of different model instances. However, exact computation of the LL is computationally difficult since a full

[2] Our implementation can be downloaded from `github.com/taygunk/robust_gram_embeddings`

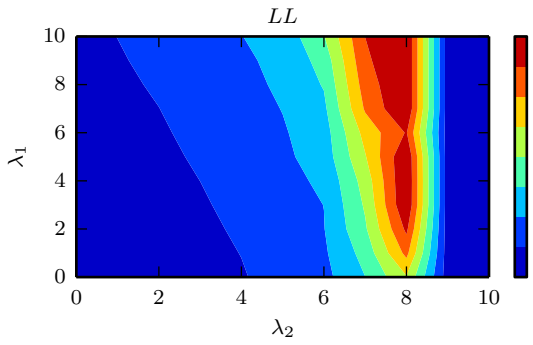


Figure 3.1: The LL objective for varying λ_1, λ_2. There exists a global minimum in the optimization landscape.

pass over the validation likelihood is time-consuming with millions of samples. Hence, we compute a stochastic likelihood with a few approximation steps. We first subsample a million samples rather than a full evaluation set, and then sample few words to predict in the window context similar to the approach followed in [19]. Lastly, we approximate the normalization factor with one negative sample for each prediction score [95][94]. Such an approximation works fine and gives smooth error curves. The reported likelihoods are computed by averaging over 5-fold cross validation sets.

Results. Figure 3.1 shows the likelihood LL obtained by varying $\{\lambda_1, \lambda_2\}$. The plot shows that there exits a unique minimum and both constraints contribute to achieve a better likelihood compared to their unregularized counterparts (for which $\lambda_1 = \lambda_2 = 0$). In particular, the regularization imposed by the differential of context and target embeddings g_2 contributes more than the regularization on the embeddings Ψ and Φ separately. This is to be expected as g_2 also incorporates an amount of norm bound on the vectors. The region where both constraints are employed gives the best results. Observe that we can simply enhance the effect of g_2 by adding a small amount of bounded norm g_1 constraint in a stable manner. Doing this with pure g_2 is risky because it is much more sensitive to the selection of λ_2. These results suggest that the convex combination of stable nature of g_1 with potent regularizer of g_2, finally yields comparably better regularization.

3.4.2. SENSITIVITY ANALYSIS

In order to test the sensitivity of our model and baseline Skip Gram to variations in the training set, we perform two sensitivity analyses. First, we simulate a missing data effect by randomly dropping out $\gamma \in [0, 20]$ percent of the training set. Under such a setting, robust models are expected to be effected less from the inherent variation. As an addition, we inspect the effect of varying the minimum cut-off parameter to measure the sensitivity. In this experiment, from a classification problem perspective, each instance is a sub-task with different number of classes (words) to predict. Instances with small cut-off introduce classification tasks with very few training samples. This cut-off choice varies in different studies [57, 93], and it is usually chosen based on heuristic and storage considerations.

Results. Figure 3.2 illustrates the likelihood of the Robust and Skip Gram model by

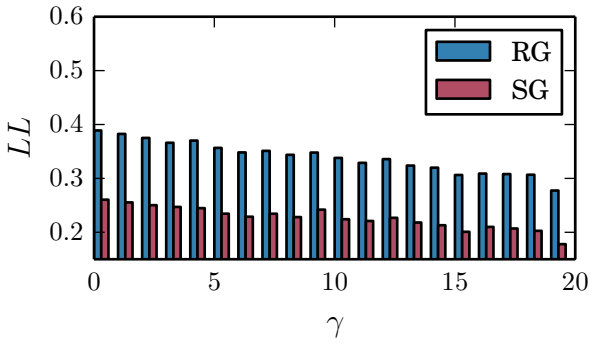


Figure 3.2: Training dropouts effect on *LL*. Our approach consistently has better Log Likelihood on γ spectrum.

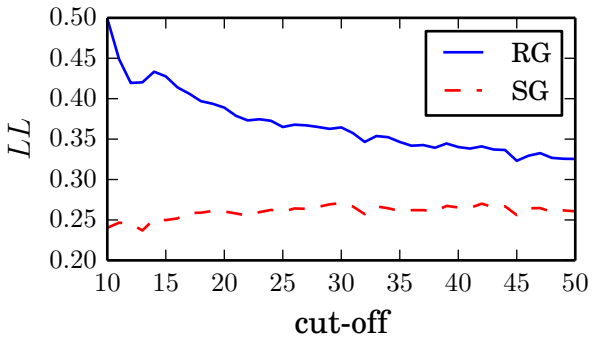


Figure 3.3: *LL* w.r.t the cut-off parameter. Increasing values effect RG's performance significantly.

varying the dropout ratio on the training set. As the training set shrinks, both models get lower *LL*. Nevertheless, likelihood decay of Skip Gram is relatively faster. When 20% drop is applied, the *LL* drops to 74% in the SG model. On the other hand the RG model not only starts with a much higher *LL*, the drop is also to 75.5%, suggesting that RG objective is more resistant to random variations in the training data.

Figure 3.3 shows the results of varying the rare-words cut-off threshold. We observe that the likelihood obtained by the Skip Gram is consistently lower than that of the Robust Gram. The graph shows that throwing out these rare words helps the objective of SG slightly. But for the Robust Gram removing the rare words actually means a significant decrease in useful information, and the performance starts to degrade towards the SG performance. RG avoids the overfitting occurring in SG, but still extracts useful information to improve the generalization.

3.4.3. WORD SIMILARITY PERFORMANCE

The work of [96] demonstrates that intrinsic tasks are a better proxy for measuring the generic quality than extrinsic evaluations. Motivated by this observation, we follow the experimental setup of [96, 97], and compare word correlation estimates of each model to human estimated similarities with Spearman's correlation coefficient. The evaluation is performed on several publicly available word similarity datasets having different sizes.

Table 3.1: Spearman's ρ coefficient for a set of benchmarks models (rows), and word similarity datasets (columns). Higher numbers are better.

	RG65	WS	WSS	WSR	MEN	RW
Size	63	353	203	252	3000	2034
CBoW	48.5	59.7	71.8	61.3	56.5	26.4
GloVe	48.9	56.2	61.5	59.1	53.0	30.0
SG	**59.2**	**71.7**	74.6	66.5	64.7	33.5
HPCA	32.1	48.6	52.9	51.5	49.9	30.7
RG	59.0	**71.7**	**74.8**	**66.7**	**65.8**	**34.0**

For datasets having multiple subjects annotating the word similarity, we compute the average similarity score from all subjects.

We compare our approach to set of techniques on the horizon of spectral to window based approaches. A fully spectral approach, HPCA, [98] extracts word embeddings by running a Hellinger PCA on the cooccurrence matrix. For this method, context vocabulary upper and lower bound parameters are set to $\{1, 10^{-5}\}$, as promoted by its author. GLoVe [93] approach formulates a weighted least squares problem to combine global statistics of cooccurence and efficiency of window-based approaches. Its objective can be interpreted as an alternative to the cross-entropy loss of Robust Gram. The x_{max}, α values of the GLoVe objective is by default set to $100, 3/4$. Finally, we also compare to shallow representation learning networks such as Skip Gram and Continuous Bag of Words (CBoW) [99], competitive state of the art window based baselines.

We set equal window size for all these models, and iterate three epochs over the training set. To yield more generality, all results obtained with 300 dimensional embeddings and subsampling parameters are set to 0. For Robust Gram approach, we have set $\lambda_1, \lambda_2 = \{0.3, 0.3\}$. To obtain the similarity results, we use the final Φ context embeddings.

Results. Table 3.1 depicts the results. The first observation is that in this setting, obtaining word similarity using HPCA and GLoVe methods are suboptimal. Frankly, we can conjecture that this scarce data regime is not in the favor of the spectral methods such as HPCA. Its poor performance can be attributed to its pure geometric reconstruction formulation, which runs into difficulties by the amount of inherent noise. Compared to these, CBoW's performance is moderate except in the RW dataset where it performs the worst. Secondly, the performance of the SG is relatively better compared to these approaches. Surprisingly, under this small data setting, RG outperforms all of its competitors in all datasets except for RG65, a tiny dataset of 63 words containing very common words. It is admissible that RG sacrifices a bit in order to generalize to a large variety of words. Note that it especially wins by a margin in MEN and Rare Words (RW) datasets, having the largest number of similarity query samples. As the number of query samples increases, RG embeddings' similarity modeling accuracy becomes clearly perceptible. The promising result Robust Gram achieves in RW dataset also sheds light on why CBoW performed worst on RW: CBOW overfits rapidly confirming the recent studies on the stability of CBoW [100]. Finally, these word similarity results suggest that RG embeddings can yield much more generality under data scarcity.

3.5. CONCLUSION

This paper presents a regularized word embedding approach, called Robust Gram. In this approach, the model complexity is penalized by suppressing deviations between the embedding spaces of the target and context words. Various experimental results show that RG maintains a robust behaviour under small sample size situations, sample perturbations and it reaches a higher word similarity performance compared to its competitors. The gain from Robust Gram increases notably as diverse test sets are used to measure the word similarity performance.

In future work, by taking advantage of the promising results of Robust Gram, we intend to explore the model's behaviour in various settings. In particular, we plan to model various corpora, i.e. predictive modeling of sequentially arriving network packages. Another future direction might be encoding available domain knowledge by additional regularization terms, for instance, knowledge on synonyms can be used to reduce the degrees of freedom of the optimization. We also plan to enhance the underlying optimization by designing Elastic constraints [101] specialized for word embeddings.

4

Semantic Vector Specializations with Bidirectional Constraint Propagations

4.1. Abstract

Word embeddings learn a vector representation of words, which can then be utilized in a large number of natural language processing applications. Learning these vectors shares the drawback of unsupervised learning: learned representations are not specialized for semantic tasks. In this work, we propose a joint formulation to effectively learn semantically specialized word vectors (Sem2Vec) by creating shared representations of online lexical sources, and formulating them as constraints to learning semantic specialization embeddings. Our results suggest that embeddings of our joint formulation are more stable and robust to variations. Further, we perform an empirical evaluation of our model on the word similarity task comprised of eleven word similarity datasets, and obtain significant boosts over state of the art competitors.

4.2. INTRODUCTION

Developing accurate representations for the word meanings is a challenging problem. Ongoing debates on the linguistics deals with the philosophical aspects of how word meanings arise for a language. In a nutshell, the widely accepted idea in the domain is the Distributional Hypothesis [102] which claims that words mostly acquire their meanings through the context they are being used. Grounding on this hypothesis allows one to develop unsupervised learning techniques to effectively learn the low order cooccurence statistics of a language. Vector space learning approaches (a.k.a. word embeddings) [13] are techniques whose representations of words are optimized such that these words and their context words are located nearby in the embedding. It appears that the resulting word vectors are usable for a diverse set of natural language applications. Recent studies have shown that these vectors yield substantial representation power and proven to be much more useful in many lingual tasks than their traditional counting based N-Gram representations [84].

Training of embedding vectors is usually performed on large unstructured corpora. A word embedding algorithm is expected to learn the structures and regularities in the language without any further guidance. Unfortunately, these algorithms share the common drawback of unsupervised learning: learned embeddings are not necessarily specialized enough for the given predictive task. Generally speaking, when one wishes to specialize the vectors for a semantic task of interest, the Distributional Hypothesis yields to be insufficient. Words occurring in similar contexts may exhibit weak or no semantic relevance, and the learned vectors do not necessarily encode features that capture semantic similarities [103].

Many formulations have been proposed to tackle this tedious and error-prone process. Incorporation of knowledge graphs [104] to the embedding network, augmenting the objective with extra relatedness annotations [105], and extraction of word senses from lexical dictionaries [106] are solutions to embed these general purpose vectors to a semantic space. The work in [107] constructs an unsupervised random field over the semantic associations to retrofit (post-process) the word vectors. These works jointly learn embeddings, given a knowledge source, and they show improvements over unsupervised, raw embeddings. Nevertheless, utilization of semantic sources is not straightforward. Each semantic source has a degree of semantic relevance to the task, and usually sources with high semantic relevance have scarce amount of data. Carefully addressing these points require the design of novel unsupervised objectives that exploits auxiliary semantic content.

In this work, we propose a novel approach to address aforementioned issues and effectively learn embeddings with semantic specializations. Briefly, the main contributions of this chapter are listed as follows:

- Each lexical source exhibits different degrees of semantic relevance. We create a shared representation of Thesaurus and online dictionaries, and then fuse these semantic content into the learning process as hard and soft constraints to restrict the original embedding problem.

- We introduce bidirectional propagations over constraint sets where 1) bottom to top augmentation propagations increase the number of hard constraints 2) top to

bottom propagations improve the overall reliability of soft constraints. This strategy constructs a well-behaving objective for learning semantically specialized embeddings. The constraint construction and propagations of our pipeline is visually illustrated in Figure 4.1.

- It is difficult to train embeddings that take long range dependencies into account. These embeddings are known to be highly unstable when trained under large window sizes. We found out that embeddings trained with our semantic constraints favors stabilized solutions almost under all query sets compared to the original embedding problem.
- Our empirical findings suggest significant improvements on semantic task evaluations. More precisely, we measure the word similarity performance of a various set of word embedding baselines using a diverse test collection comprised of eleven datasets. Our weighted average of Spearman correlation scores, yield a 4.3% improvement upon the state of the art solutions. The improvement over the competitors is much more significant with a 7.4% when embeddings are trained on a smaller subset of Wikipedia 2017.

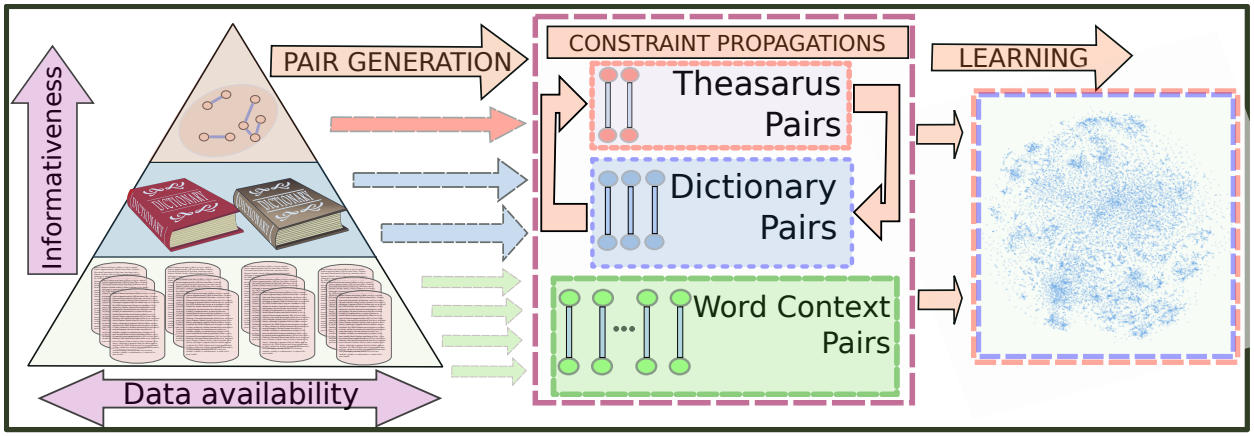


Figure 4.1: Our proposed word embedding pipeline (best viewed in color). We first generate various levels of word context pairs using a triplet of sources: unsupervised corpora, lexical dictionaries and Thesaurus. We then treat upper lexical sources as optimization constraints and perform bidirectional propagations between the constraint sets to maximize the learning efficiency. Our final word embeddings are highly suited for semantic tasks.

4.3. PROPOSED APPROACH

In this section, we introduce the preliminary word-context learning problem, followed by construction of our hard and soft constraints. We then conclude by detailing our bidirectional constraint propagations.

4.3.1. WORD VECTOR MODELS

A large set of word embedding approaches use the following objective function:

$$J(w,c) = \ell(w,c) - \sum_{c_N \in V_c} \ell(w,c_N) \tag{4.1}$$

where w is a target word in vocabulary V_w, and c is a context word in a vocabulary set V_c. Further we define $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$ as the vector representations of w and c, respectively. Then $\ell(w,c) = \log\left(1+\exp(-\vec{\boldsymbol{w}}^T\vec{\boldsymbol{c}})\right)$ is the logistic loss function. The first logistic loss term in the objective penalizes the dissimilarity of $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$. The second term is the negative sampling contribution, in which some randomly sampled contexts c_N are forced to have a vector $\vec{\boldsymbol{c}}_N$ that is most dissimilar to $\vec{\boldsymbol{w}}$. The total loss over the corpus is then simply the sum of i.i.d. (w,c) word context pairs. This objective is an application of distributional hypothesis: if a word w occurs together with context c, they should have similar vectors. This relation is weighted more if they cooccur more in the observed corpus.

4.3.2. SEMANTIC WORD VECTOR SPECIALIZATIONS

The learning embeddings using Equation 4.1 attained reasonable success for the general tasks. However, when we want to specialize embeddings for semantic relations, we notice several problems with this approach. First, given a word, its semantic partner (e.g. its hypernym, hyponym or synonym) usually occurs with it usually only through long range dependencies [108]. It is very unlikely to observe a word and its semantic partner together in a local window.

The second difficulty is that unsupervised objective has no preference over any pairs. Without explicitly telling the model which loss pairs are semantically valuable, most of the loss pairs are those that do not necessarily have a strong semantic informativeness. For instance, consider the sentence: *"my dog is a nice and big one, and like to eat high quality food"*. According to the Distributional Hypothesis; the meaning of *dog* and stop words like *a, is, one* should be closer to *dog*, although we know the semantic relation here between *dog* and *food* is much stronger.

These two problems arising from the hypothesis can be addressed by guiding the objective function, such that it weights semantically valuable pairs heavier than the rest. This is possible by leveraging auxiliary semantic information that specify the feasible regions of the objective function in Equation 4.1. But which pairs are more semantically valuable? From a computational linguistics point of view [109], the value of semantics is understood via the concept of *Information Content* which suggests that general entities present less information than the more specialized entities and relations. In other words, abstract relations of semantics have high information content whereas raw cooccurences provide significantly less amount of semantic content. Consider the relations of two words w and c. These words can cooccur in a domain such as raw noisy corpora, a dictionary, or in a thesaurus. As information content suggests, these relations differ in their semantic abstraction level: there is a clear distinction between the raw text cooccurence relation and a dictionary sense relation, the latter indicating a stronger relation.

Lexical Dictionary. The lexical dictionary is a rich source containing sense definitions of the words where one can extract significant clues what the meaning of the word is with respect to other words. For example, consider the definition of word *tower* in Table 4.1. There are commonalities across the definitions of the same word. For our purposes, we extract all word-context pairs from the dictionary definitions, and denote an extracted elements as *sense pairs*. Let **D** be the dictionary. We formulate a sense pair as a constraint to the semantic similarity of (w,c). We penalize the dissimilarity of $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$ under the logistic loss, and form a constraint $\ell(w,c) \leq \tau$. Then for any word-context pair

Table 4.1: Dictionary and Thesaurus content for the query word *tower*.

Source	Content
Dict [1]	*a building or structure high in proportion to its lateral dimensions, either isolated or forming part of a building.*
Dict [2]	*A tower is a tall, narrow building, that either stands alone or forms part of another building such as a church or castle.*
Theasarus [3]	*belfry - castle - citadel - column - fort - fortification - fortress - keep - lookout ...*

[1] http://www.dictionary.com [2] http://en.oxforddictionaries.com [3] http://www.thesaurus.com

in the learning problem, we have the following soft constraint via the dictionary:

$$\big(\ell(w,c) \leq \tau\big)\, \mathbb{1}_{\mathbf{D}}(w,c)$$

where

$$\mathbb{1}_{\mathbf{D}}(w,c) = \begin{cases} 1 & (w,c) \in \mathbf{D} \\ 0 & otherwise \end{cases}$$

is the indicator function for the cooccurence of (w,c) in dictionary set $\mathbf{D}$. We then use standard Karush Kuhn Tucker (KKT) conditions [110] to treat this dictionary constraint as an objective term:

$$J_{\mathbf{D}}(w,c) = \lambda_{\mathbf{D}} \big(\ell(w,c)\, \mathbb{1}_{\mathbf{D}}(w,c)\big) \tag{4.2}$$

where τ disappeared since it neither depends on w nor c. Since dictionary pairs are considered as a constraint to raw cooccurences: we call $J_{\mathbf{D}}(w,c)$ as the *soft constraint* objective.

Thesaurus. Thesaurus is a reference source where a word is explained in a concise manner using a small subset of vocabulary words. In contrast to a dictionary, thesaurus does not treat words in an alphabetic order. Also, dictionary definitions can contain syntactic or semantic relevance, yet Thesaurus only accounts for semantic relations. These relations are very abstract and may contain synonyms and antonyms. The pure semantic nature of the Thesaurus means that pairs generated from it have higher information content than dictionary pairs. For the word *tower* the last row of Table 4.1 shows the query result from a thesaurus. We see that the Thesaurus definition of *tower* is much condensed compared to dictionary content, and mostly includes concrete building objects having structural similarities.

Similarly to the dictionary definitions we extract pairs, and denote $\mathbf{T}$ as the set of Thesaurus pairs to further constrain the embedding problem. That is we penalize the dissimilarity under the logistic loss and form the hard constraint through the Thesaurus:

$$\big(\ell(w,c) \leq \tau\big)\, \mathbb{1}_{\mathbf{T}}(w,c)$$

where $\mathbb{1}_{\mathbf{T}}(w,c)$ is the indicator function for the Theasarus $\mathbf{T}$ for the pair. This hard constraint is converted to an objective term:

$$J_{\mathbf{T}}(w,c) = \lambda_{\mathbf{T}} \big(\ell(w,c)\, \mathbb{1}_{\mathbf{T}}(w,c)\big) \tag{4.3}$$

where $J_{\mathbf{T}}$ is the objective contribution from the Theasarus. In here *hard* means the constraint has to be strictly satisfied during the optimization problem. This is characterized by using $\lambda_{\mathbf{T}}$ such that $\lambda_{\mathbf{T}} \geq \lambda_{\mathbf{D}}$ holds.

4.3.3. BIDIRECTIONAL CONSTRAINT PROPAGATIONS

In the last subsection, we constructed soft constraints from the lexical dictionary and hard constraints from a Thesaurus source. These constraints restrict the maximization of the objective function over a subspace that semantic relations hold. Unfortunately, sets with high information content are very much limited in size as Figure 4.1 demonstrates. On the other hand, dictionary pairs are relatively less informative but potentially can yield to an order of magnitude more constraints. The main idea in this subsection is that these two lexical sources can mutually benefit from each other. Promoting reliable sense pairs can increase the number of hard constraints and Thesaurus can create some new constraints for the dictionary to increase its average informativeness.

For promoting a soft constraint to hard, we define two rules:

- *definitional symmetry.* The dictionary sense definition pair is denoted as symmetric if $(w, c) \in \mathbf{D}$, and $(c, w) \in \mathbf{D}$. This indicates a very strong semantic relation, and we promote this pair to be an element of $\mathbf{T}$.

- *expert agreement.* Assume we have d dictionaries collected from independent sources representing our large dictionary set $\mathbf{D} = \{\mathbf{D}_1, \mathbf{D}_2, ...\mathbf{D}_d\}$. If the definition of word w contains c in multiple dictionaries, then (w, c) pair is an expert sense. According to this rule, the word *tower* in Table 4.1 has *building* in its definition across multiple dictionaries. Hence, *tower-building* is an expert agreed sense. We augment $\mathbf{T}$ with these pairs.

In the next step, we query elements of $\mathbf{T}$ and stochastically apply semantic association rules to form new pairs. While there exists ontology knowledge based association rule techniques [111], we adopt a low complexity association rule that is if a pair (w_1, c) are (w_2, c) both in $\mathbf{T}$, we then create (w_1, w_2) pair and add it to the set $\mathbf{D}$. We perform these associations for a tiny number of KNN neighbourhood and increase the average information content of the soft constraint set.

4.3.4. LEARNING BY CONTROLLED NEGATIVE SAMPLING

A common technique to learn word embeddings is the negative sampling [112]. In this approach, a noise distribution generates word context pairs and the model is trying to learn by discriminating between positive word-context pairs and negative pairs. Negative sampling contribution term is:

$$J_N(w) = \sum_{c_N \in V_c} \ell(w, c_N)$$

When using this approach, we must ensure that w and c_N are not related. In our approach we know pairs obtained from $\mathbf{T}$ and $\mathbf{D}$ are strongly related. There is still a non-zero probability to sample such pair. To overcome this issue, we perform Controlled

Negative Sampling similarly to [106]. We do not negative sample if the pair is in these sets:

$$J_N(w) = \sum_{\substack{c_N \in V_c \\ (w,c_N) \notin \mathbf{T} \\ (w,c_N) \notin \mathbf{D}}} \ell(w, c_N) \tag{4.4}$$

This discards a small fraction of the negative samples from the objective and yields better learning. Our final objective is the sum of the pair loss, J_N, J_T and J_D:

$$J(w,c) = \ell(w,c) + J_T(w,c) + J_D(w,c) - J_N(w) \tag{4.5}$$

where the global objective can be easily obtained by simply summing over all i.i.d (w,c) pairs in the corpus.

4.4. EXPERIMENTAL RESULTS

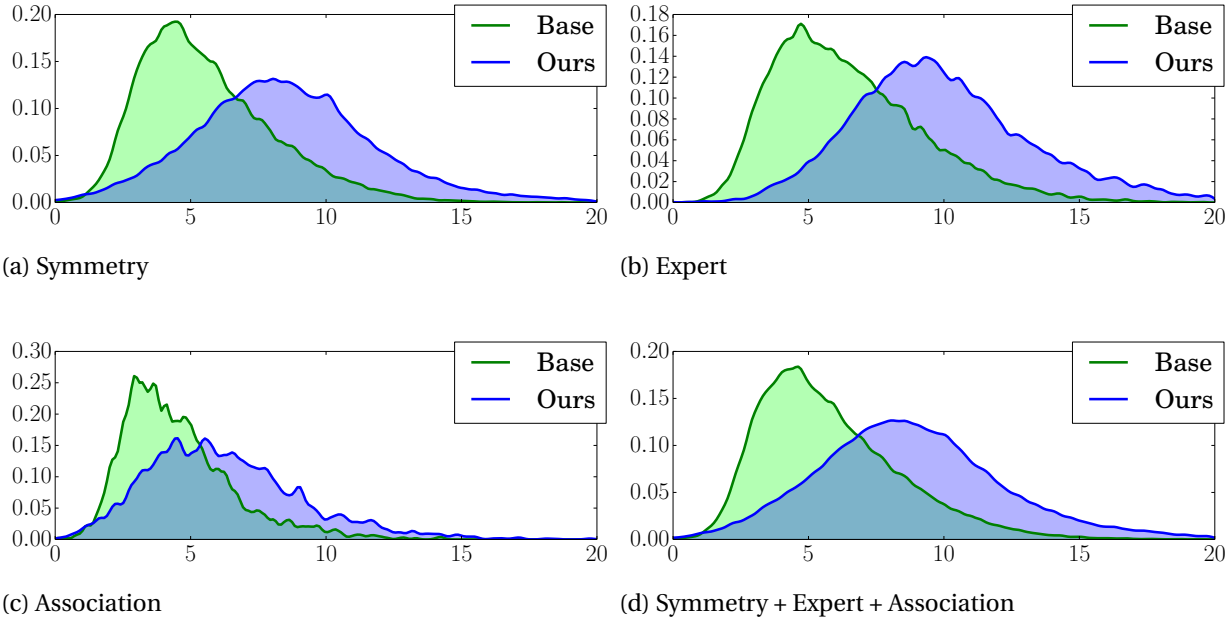
Experimental Setup. We train our embedding models using the latest Wikipedia 2017 July snapshot containing 4.5B tokens. We extract the vocabulary from the corpus which gives us approximately a vocabulary of 2.3M words. Our corpus processing follows the state of the art practices for Wikipedia which we use the standard preprocessing scripts and remove XML and HTML tags to obtain the raw text [99]. For a fair evaluation, all common embedding training parameters are set as in [113], where we remove words that occur less than 5 times, set the window size to 5, number of negative samples to 5, and corpus is processed for 5 epochs. The initial learning rate is set to same value for the methods and Stochastic Gradient Descent is used as the optimization algorithm.

Dictionary and Thesaurus Collections. We use Cambridge, Oxford, Collins, Dictionary.com and Longman English dictionaries to obtain word definitions. Similarly to [106], we crawl the dictionaries with web requests and parse the HTML contents using regular expressions to get word definitions from Cambridge, Oxford, Collins, Dictionary.com. Unlike other dictionaries, the Longman Dictionary provides an Application Programming Interface ,Longman Pearson API, allowing to directly get the word definitions. The definition texts are preprocessed similarly to the input corpus such that only alphanumeric characters are present. For obtaining more accurate pairs, we reduce the redundancy by removing the stop-words from dictionary definitions. After collection of all definitions from all dictionaries, as the purpose is not word sense disambiguation, we concatenate all senses into a single list. For a Thesaurus source, we crawl the contents of Online Thesaurus[4] where each word is provided a list of synonyms. After the initial construction of our hard and soft objective terms using pairs from our sources, we apply the bidirectional constraint propagations.

Methods. Our performance benchmarks includes comparisons with the following architectures:

- SG [99]: The vanilla baseline using Skip Gram architecture of Word2Vec. We refer to this architecture if no abbreviation is given.
- CBoW [112]: state of the art architecture representing the context vectors as the bag of words around the target word. This architecture is faster than SG.

[4] http://www.thesaurus.com



(a) Symmetry

(b) Expert

(c) Association

(d) Symmetry + Expert + Association

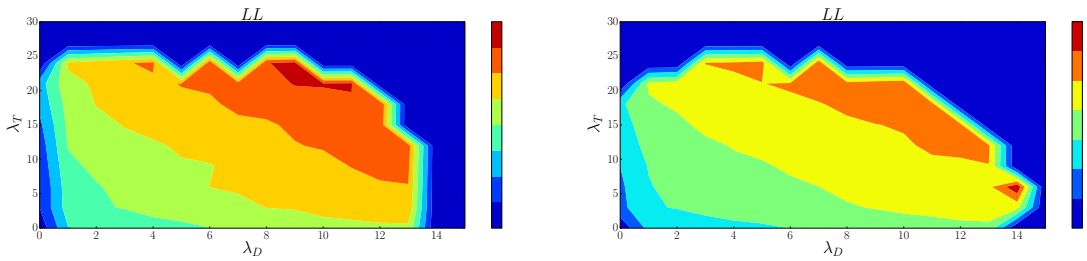
Figure 4.2: Kernel Density Estimate fits to inner products. x-axis is the inner product value and y-axis is the density estimate. a) symmetry pairs b) expert pairs and c) association pairs. In d) we show the inner product density for all pairs.

- Dict2Vec [106]: embedding architecture using dictionary definitions. As their approach requires a preliminary training step of word embeddings, we first pretrain the embeddings to obtain initial vectors. We then follow identical steps: use pretrained vectors to specify and promote the constraint pairs and set parameters to the best reported results.
- FastText [113]: embedding architecture where each word is represented as a bag of character N-grams. This is one more extra layer of word representation where vectors enjoy the additional shared knowledge of N-Grams. For parameter specification, we use the default suggested settings for their bucket length, N-Gram sizes and update rates.
- Our approach. For setting our hyperparameters λ_D and λ_T, we follow the same protocols as in [106].

We also ran our experiments with GLoVe [93] and HPCA [98] embeddings but we could not obtain comparable results with these embeddings, so we do not report their results.

4.4.1. QUANTITATIVE RESULTS

Constraint Propagations. Some random pairs obtained from our bidirectional propagation step are shown in Table 4.2. Symmetry and expert agreement pairs highlights strong semantic relevancies. As there seems to be a low deviation in the conveyed meaning for some of these pairs, arguably, these can even be used as meaning-preserving substitutes for training a lexical substitution system (e.g. "examination-test", "forbidden-taboo"),



(a) No propagation

(b) with propagations

Figure 4.3: Log Likelihood landscapes for $\{\lambda_{\mathbf{T}}, \lambda_{\mathbf{D}}\}$ a) no propagation b) with constraint propagations. In both cases, there exists global maxima for the landscapes.

Unlike symmetry and expert pairs, association pairs instead depicts gripping cases. This step generates pairs like "science-aesthetic" which captures a usually omitted dimension of word science, suggesting the "science" is not only functional, but also contains an aesthetics aspect. Another pair "international-alphabet" is an example how simple associations on word pairs can also point to phrasal concepts such as the *Phonetic Alphabet*. Compared to symmetry and expert relations, these associations also generate potentially valuable semantics that we do observe in the corpus.

To measure how these word pairs are affected when we apply our model, we fit a Kernel Density Estimate to the cosine distances of pairs for symmetry, expert, association and depict the results in Figure 4.2. Satisfying our expectations, learning with our model causes all densities to undergo a mean shift and yield higher average inner product. The density shift is relatively larger in expert pairs compared to symmetry and associations, suggesting that the expert agreement has the strongest impact in our constraints. Furthermore, observe that original densities for these word pairs are right (positive) skewed. This is logical when there is no prior knowledge available for semantics, most of the pairs are tend to have low cosine similarity. Learning with our model corrects the inherent skew, and yields a Gaussian-peaked concentration for inner products.

4.4.2. MODEL SELECTION

Setting. For model selection purposes, we analyze the likelihood of multiple trained instances of our model. We form a large validation set containing millions of words and then evaluate the predictive likelihood of each model instance on this set. Since exact computation is not feasible, similarly to stochastic computations in [19], we compute a stochastic likelihood with sampling few context words around the target word and randomly perform multiple repetitions.

Results. Figure 4.3a and Figure 4.3b depicts the likelihood LL contours over the $\{\lambda_T, \lambda_D\}$ grid without and with constraint propagations. We observe on both settings landscapes exhibit a unique maximum. Constraint propagations increases the smoothness of the landscape, contour edges yield smoother transitions. This means for any optimization algorithm, it is easier to discover a better maximum when new constraints are formed using these propagations. In particular, the slope of the contours also show that hard constraints of Thesaurus is much more informative compared to the ones obtained from Dictionaries. The orientation of the contours suggest that there is a linear

Table 4.2: Some example word pairs from propagation sets.

Symmetry	Expert	Association
coal-fuel	forbidden-taboo	time-atomic
examination-test	hit-serve	abroad-disperse
gold-jewellery	crack-open	natural-harmony
carry-serve	microscobic-small	society-tandem
medicine-surgery	existence-produce	art-witchcraft
address-addressed	disrupt-prevent	black-gathering
break-disrupt	cave-hill	science-aesthetic
short-summary	pond-water	dignity-quality
box-wagon	fall-shower	international-alphabet
college-institution	cache-hidden	language-grammatical

relationship between λ_T and λ_D, that suggests the relative weighting of these sources. Under our embedding model for learning semantics, a Thesaurus is worth ten (we used dictionary constraints from aggregating 5 independent sources) dictionaries.

4.4.3. EMBEDDING STABILITY

Setting. In this section, we want to measure the stabilization effects of using our embedding technique. To be able to capture long range dependencies of word cooccurences, large window sizes have to be used [108]. Nevertheless, experimental evidence [100] shows that embeddings obtained from such training conditions are shown to be highly unstable. To understand the behaviour of the models, we simply train multiple randomly initialized embeddings and check how the nearest neighbours of the query words are subject to variations. We first train multiple random embeddings, and store the nearest neighbours of query words using cosine similarity. Then, similarly to [114] we use a stability measure based on Jaccard Index for comparing the similarity and diversity of sample sets. The index is defined as the size of the intersection divided by the size of the union of the sample sets: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ where A and B are embedding sets for a set of word queries. For query sets, we use word similarity datasets as well as the recently proposed Sch dataset of [96], that is calibrated well according to word frequencies, and also considers parts-of-speech and abstractness of words into account.

Results. Figure 4.4 depicts the mean and variance of the Jaccard Index for each query inventory. The stability significantly deteriorates on large window sizes with the typical embedding learning approach. The mean deterioration trend is mostly linear for RW and Sch datasets, and variances are comparably similar. Our approach does not deteriorate on large window sizes, instead yields increased stability. The stability results strongly suggests that learning the embeddings does possess high degrees of freedom in the optimization, maybe even more than necessary, carrying the risk of forming random neighbours for words for each training instance. Introduction of our constraint pairs serves as a stabilizer for avoiding these solutions. Since the stability index results of our approach suggests that the model is even rejecting some weak word neighbours that is slightly recommended by the corpus.

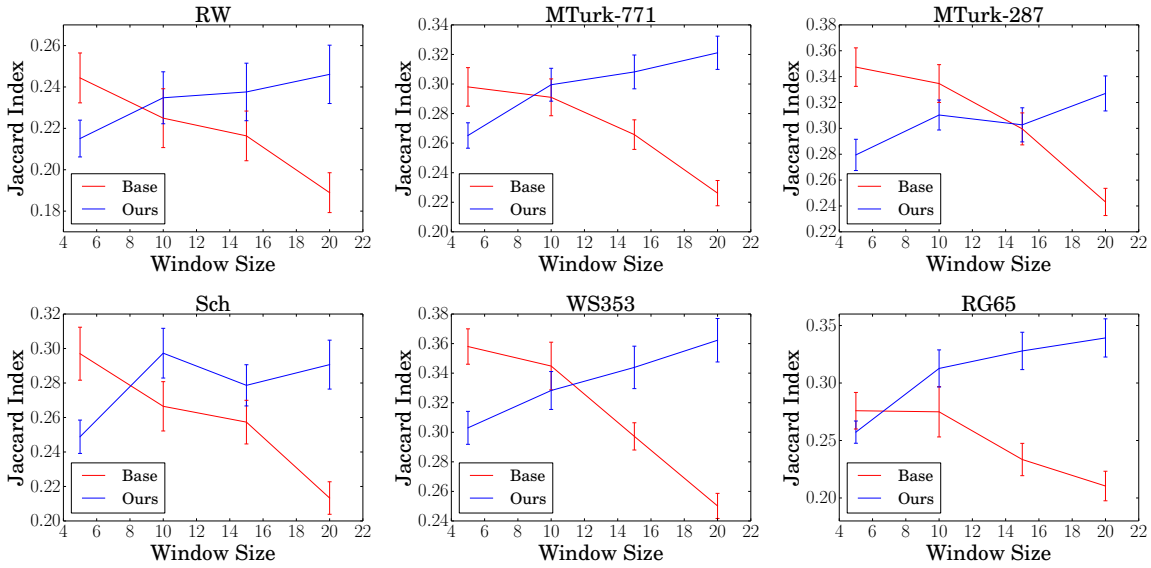


Figure 4.4: Jaccard Stability Index on different query inventories. Despite the traditional approach, the stability does not detoriate with our approach. The embeddings yields to be highly stable especially for large window sizes.

In Figure 4.5, we project the word vectors to 2D space using t-SNE dimensionality reduction [115] and show how the close proximity of a randomly sampled word ("feasible" in this case) change. Each column shows the neighbourhood of a training instance. The circle radius' for the neighbour indicates how many times it appears as the neighbour of the word in total. First row shows instances from baseline training, and the second row shows instances of our model. We observe more stationary neighbours when training includes our constraints.

Embeddings trained with our semantic constraints favours stabilized solutions for all query sets compared to the original embedding problem, and might be also utilized when the task of interest asks for large window dependency modelling.

4.4.4. WORD SIMILARITY MEASUREMENTS

Data and Parameters. We report both the similarity results for embeddings trained on the first 50M words and the 200M version of the Wikipedia. For a fair comparison against all other baselines, we also concatenate the collected definitions and synonyms to the training data so that other methods can also benefit from the extra sources. Wiki 50M denotes the raw training corpus whereas Wiki 50M+ is the corpus with pair concatenations.

Table 4.3: Word Similarity performances of embeddings trained on first 50 Million words, and 200M version of Wikipedia 2017.

	Wiki50M					Wiki50M+				
	SG	CBoW	D2V	FT	Ours	SG	CBoW	D2V	FT	Ours
MC-30	69.9	64.2	**74.5**	74.1	72.0	76.7	72.9	75.3	**78.5**	77.6
MEN	69.5	65.3	71.1	70.4	**72.1**	71.7	66.7	72.0	72.1	**72.3**
MTurk-287	65.4	65.5	66.6	66.0	**68.5**	65.6	65.3	66.6	67.6	**68.0**
MTurk-771	61.4	56.3	65.6	59.9	**70.2**	64.7	60.9	67.6	64.5	**70.9**
RG-65	70.0	67.5	76.8	69.9	**80.6**	80.3	75.3	82.0	78.0	**83.9**
RW	40.9	31.2	43.4	44.9	**49.2**	46.9	40.4	47.9	49.1	**50.9**
SimVerb	20.8	15.5	29.8	19.7	**43.5**	30.0	23.4	35.7	28.6	**47.1**
WS	69.9	62.7	**74.2**	67.2	71.6	72.2	64.1	**73.6**	68.3	72.7
WSR	64.6	55.7	**67.9**	62.9	61.5	65.6	56.3	**67.3**	63.3	63.5
WSS	75.6	68.6	77.8	72.4	**77.9**	77.8	71.1	78.0	75.2	**78.9**
YP-130	39.8	32.5	56.0	46.3	**67.5**	54.7	47.2	58.7	59.1	**67.6**
W. Average	46.9	41.1	51.7	47.4	**57.9**	52.4	46.5	54.9	52.3	**59.7**
	Wiki200M					Wiki200M+				
	SG	CBoW	D2V	FT	Ours	SG	CBoW	D2V	FT	Ours
MC-30	78.6	66.4	78.5	73.4	**79.6**	**79.3**	76.0	78.2	77.9	**79.3**
MEN	71.3	67.1	72.6	71.5	**74.5**	72.5	68.7	72.0	74.4	**75.3**
MTurk-287	65.4	65.5	64.8	**67.2**	66.5	64.0	63.9	64.2	**69.1**	66.7
MTurk-771	61.7	57.2	66.2	60.1	**73.2**	64.7	60.1	67.5	68.1	**74.2**
RG-65	74.6	70.9	79.2	69.7	**83.6**	79.1	77.5	81.2	79.9	**85.6**
RW	43.1	37.4	45.8	46.5	**53.1**	47.6	43.5	49.2	**54.5**	53.6
SimVerb	20.9	15.7	29.6	19.0	**43.6**	26.7	23.0	33.7	35.7	**46.8**
WS	70.3	62.7	72.9	67.2	**74.2**	71.0	63.1	**73.2**	71.4	**73.2**
WSR	63.9	55.8	66.2	62.1	**66.3**	64.9	56.3	65.4	**65.7**	64.6
WSS	76.4	69.6	78.2	72.1	**80.8**	76.9	70.2	78.3	76.5	**79.9**
YP-130	33.2	24.3	50.7	46.5	**68.1**	47.5	40.2	57.3	63.3	**69.2**
W. Average	47.9	42.8	52.4	47.8	**59.8**	51.5	47.4	54.4	56.9	**61.2**

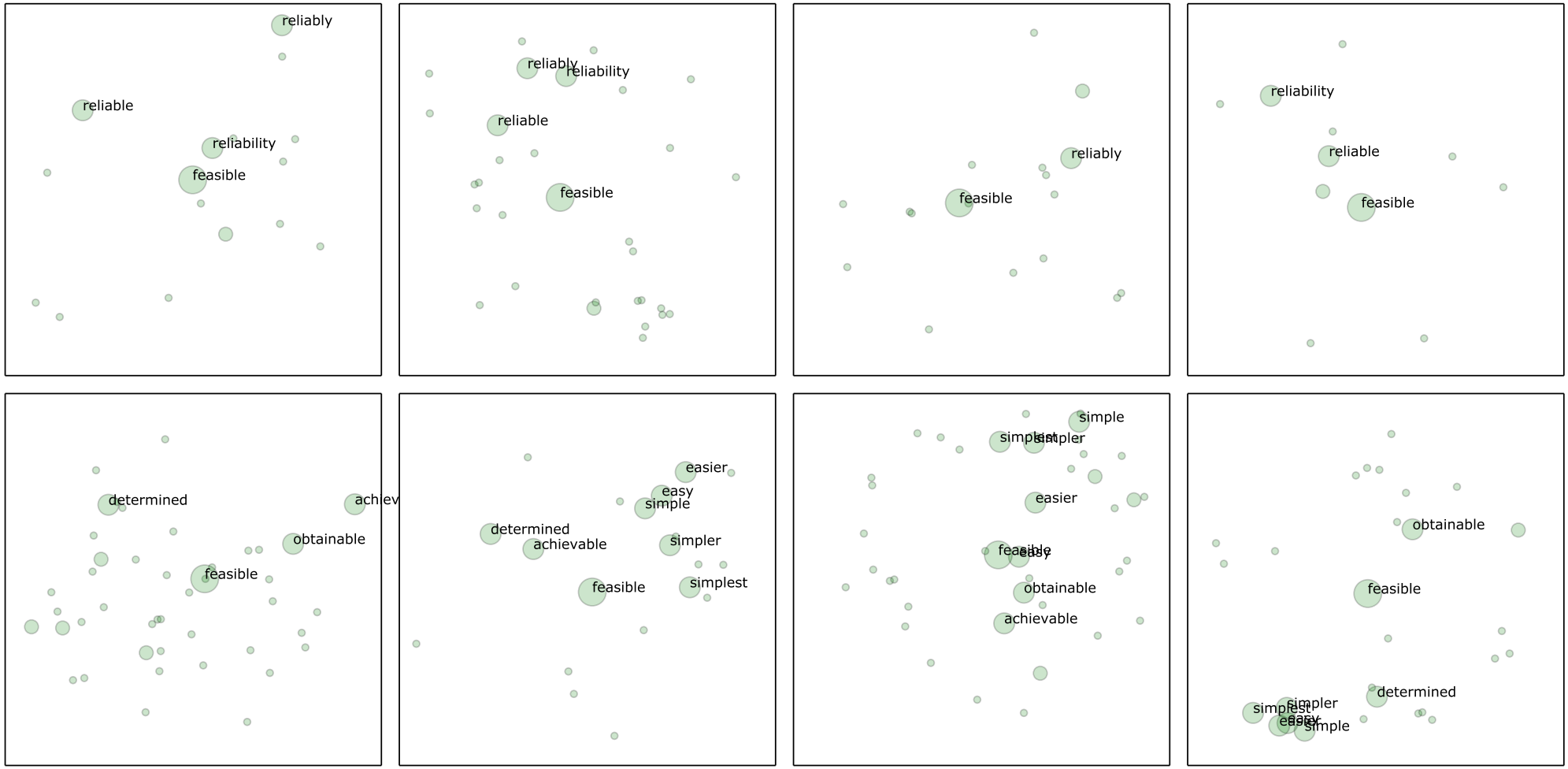


Figure 4.5: Neighbouring proximity of the word *feasible*. First row is the baseline, and second row is our approach. Each column is the resulting neighbourhood of a random training instance. The circle radius' is an indicator of how many times that word appears in all instances. Training with our constraints preserves many of the neighbouring words.

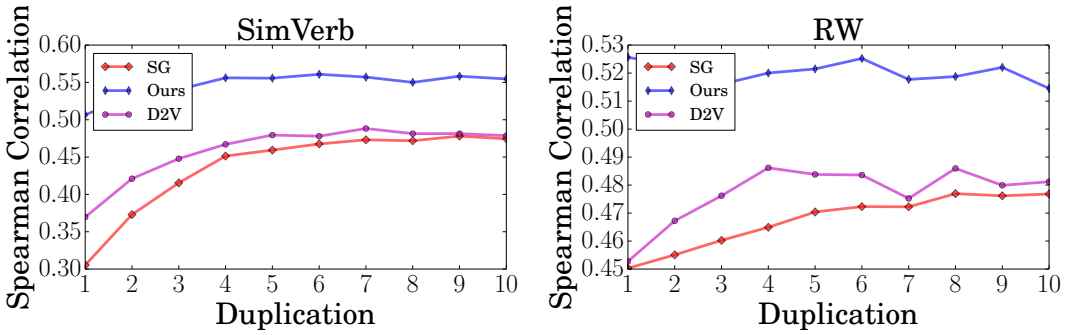


Figure 4.6: Word Similarity performances when semantic sources are concatenated multiple times to the training corpus. The gain for other embedding architectures quickly saturates.

4

Evaluation and Baselines. We test our embeddings on a large set of test collections. As a standard extrinsic benchmark of [96], we compute the Spearman Correlation Coefficient of cosine distances of word pairs to measure how much embeddings can predict the expert annotated similarities. We measure the weighted average by weighting each dataset by its query inventory size. We test our embeddings on: MC-30 [116], MEN [117], MTurk-287 [118], MTurk-771 [119], RG65 [120], RW [121], SimVerb-3500 [122], WordSim-353 [123] and YP-130 [124]. To increase the confidence of the experiments, we repeat each experiment with different seeds and report the averages.

Quantitative results. Results are reported in Table 4.3. For models trained on Wiki50M corpus, the gain of our approach over FastText reaches 10.5%, and Dict2Vec by 6.2% on dataset average. On dataset basis, our method obtains highest gains for SimVerb and YP-130 datasets. For models trained on the concatenated Wiki50M+ corpus, other methods yield a 4.75% increased performance, whereas our model obtains 1.2% extra on the Wiki50M corpus, as it already learned embeddings on the constrained subspace. Concatenation of pairs from the semantic sources as training input can benefit all models only for a few percents. The contribution is largest for the RW and Simverb datasets. Here, SimVerb contains many pairs for the syntactic and semantic similarities of verb meanings. RW dataset contains query pairs that are observed only few times in the corpus. We understand that leveraging pairwise constraints helps most for learning the verb meanings, and also for out of vocabulary queries. Our observations are similar when training on the 200M version, except that a few percents extra performance is obtained, with FastText gaining the most from the concatenation routine.

Sample duplication results. It is also a question of interest whether we can treat the semantic sources as samples, and apply sample duplications rather than extending the formulation with constraints. We demonstrate the consequences of this scenario in Figure 4.6 where we simply extract all pairs from sources and concatenate them multiple times to the available corpus. The first few duplications raises the performance greatly, but gain saturates around 10 duplications where no extra benefit is observed. For our approach, duplications only cause small fluctuations in the word similarity performance. It turns out that duplications are an alternative approach to embed semantic knowledge to the learning problem while introducing little extra training time, nevertheless the performance gain is far away from optimal.

Economical scenario. So far we assumed that the model has access to a highest level

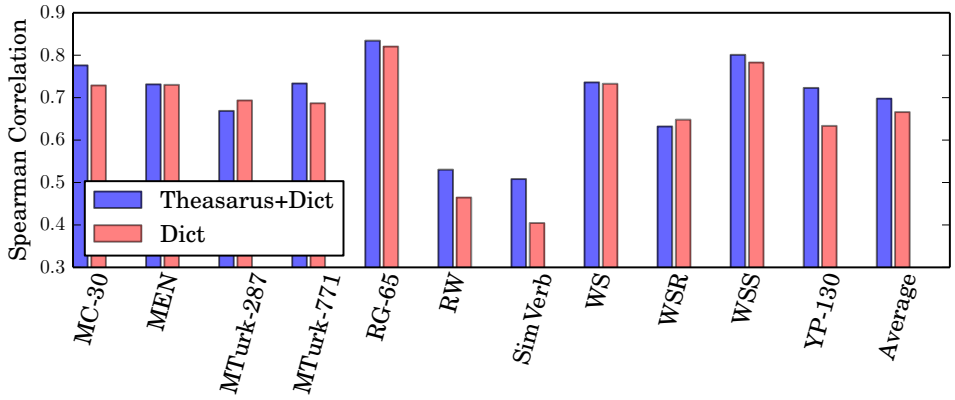


Figure 4.7: Word similarity performance when high level semantic source is unavailable.

semantic source during training. Under some conditions, this assumption might be too optimistic since for many languages these semantic sources might not be either available or accessible. We name this condition as an *economical* one for the word embedding learning. In Figure 4.7, we demonstrate how word similarity performance varies when we are only left with a dictionary source and lose access to the Thesaurus content. On all datasets, losing access to the Thesaurus harms the performance. We observe significant performance losses on the RW and SimVerb datasets. Comparably, the drop is less significant for easy datasets containing very frequent words such as RG-65 and WSS. This suggests that learning word similarities can be done using only lexical dictionaries, given that test sets query relatively easy word pairs. On the other hand, if test sets contain pairs that are rare, exploiting a higher level of semantic source appears to be indispensable.

4.5. CONCLUSION

In this work, we proposed a novel embedding framework to learn vectors specializing to semantics. Our word embedding pipeline integrated various levels of semantic sources into one unified formulation by treating highly confident lexical sources as hard constraints, and lexical dictionaries as soft constraints to learning. We then utilized the domain knowledge inherent in the lexical sources to further refine our constraint sets by bidirectional propagations, yielding a better behaving objective function.

Our constrained embedding formulation is found out to be more stable than typical word embeddings, especially for training settings on the large window sizes. We empirically evaluated how much gain our model implies for word similarity measurements, suggesting significant boosts performance over multiple baselines. Furthermore, the limitations of sample duplications as integrating semantic knowledge to the embeddings is highlighted and compared to our constraint based formulation. Worst-case economical scenarios in which a semantic source is unavailable is investigated and performance losses are discussed. Practical contribution of our model on the word similarity test suite of eleven datasets is measured, showing significant performance improvements over the state of the art techniques.

Perhaps a notable merit of our formulation is that it integrates semantic knowledge to the features but follows the conventional word embedding pipeline where training

does not require any human intervention. This is an important remark to obtain vectors in a manageable time since most of the embedding architectures require a human in the loop, significantly slowing down the training procedure. We conclude that state of the art vectors do not have any guarantee to learn semantic relevancies especially when the amount of training data is scarce for a given language in which Sem2Vec embedding approach provides a not only stable but also time-efficient embeddings to learn these semantic relevancies.

5

BOOSTED NEGATIVE SAMPLING BY QUADRATICALLY CONSTRAINED ENTROPY MAXIMIZATION

5.1. ABSTRACT

Learning probability densities for natural language representations is a difficult problem because language is inherently sparse and high-dimensional. Negative sampling is a popular and effective way to avoid intractable maximum likelihood problems, but it requires correct specification of the sampling distribution. Previous state of the art methods rely on heuristic distributions that appear to do well in practice. In this work, we define conditions for optimal sampling distributions and demonstrate how to approximate them using *Quadratically Constrained Entropy Maximization* (QCEM). Our analysis shows that state of the art heuristics are restrictive approximations to our proposed framework. To demonstrate the merits of our formulation, we apply QCEM to matching synthetic exponential family distributions and to finding high dimensional word embedding vectors for English. We are able to achieve faster inference on synthetic experiments and improve the correlation on semantic similarity evaluations on the Rare Words dataset by 4.8%.

5.2. INTRODUCTION

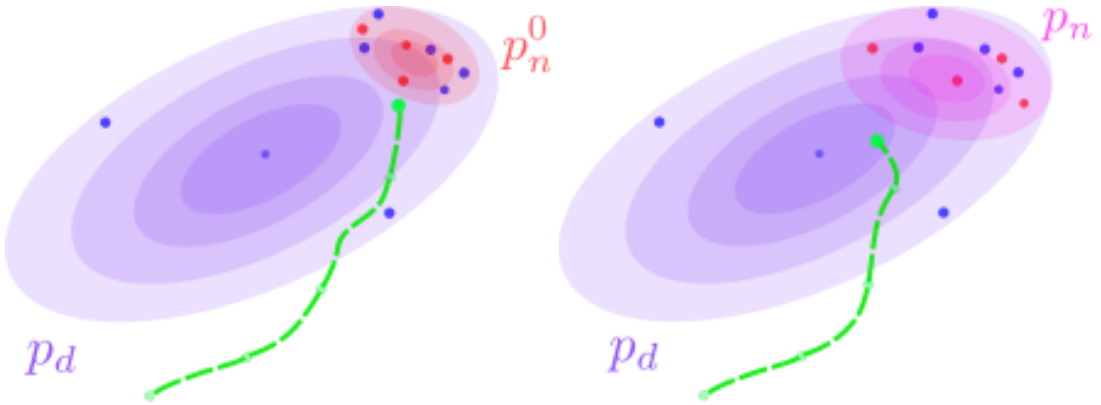
The combination of large, publicly available text collections and distributed word vector representations [13] has revolutionized our ability to study the underlying structural patterns of language. Distributed representations, or word embeddings, operationalize the distributional hypothesis [125], which asserts that words acquire meaning over time through their contexts. Embeddings approximate these contextual meanings by mapping words to continuous vectors, so that words that occur in similar contexts have similar vectors.

Recently, studies have shown that these vectors yield substantial representation power and proven to be much more useful in many lingual tasks than their traditional counting based N-Gram representations [84]. Nowadays, word embeddings are typically adopted as fundamental building blocks for a variable set of linguistic tasks [126]. Some successful applications of such vectors are sentiment classification [127], sarcasm detection [85], question answering [128], cross-language text classification [129], recommendation systems [130].

Word embeddings are typically dense and have radically lower dimensionality than the number of words in a language, but they are nevertheless still high dimensional. Traditional statistical estimators such as Maximum Likelihood Estimation (MLE) easily becomes intractable for learning these high dimensional models [131]. Negative sampling on the other hand, derived from the contrastive learning, easily scales up to large embedding models. Although scalability is an attractive property itself, the user still has to consider design issues to ensure successful learning with negative sampling. Since we have limited data in many practical word embedding problems, it becomes crucial to use a reasonable sampling distribution in order to fit accurate models.

In this work, we address the aforementioned problems of word embedding architectures using negative sampling as the learning component. To achieve this, we propose a relaxed Maximum Entropy based sampling principle. Main contributions of this chapter can be summarized as follows:

- An objective is obtained which expresses the effect of a sampling distribution with a physical analogy, as attractive and repulsive forces. This formulation lends to a Maximum Entropy formulation.
- A surrogate smoothing objective to the original problem: Quadratically Constrained Entropy Maximization (QCEM) is proposed, posing a computationally attractable method for choosing sampling distributions. Our proofs show that state of the art heuristics are simple and restricted approximations of our general maximization framework.
- Empirical findings on learning synthetic exponential family densities are provided for analysing the convergence rates of methods.
- The merits of our approach are further demonstrated on word vector space learning when data is scarce and limited. We report word similarity performances on a large number of datasets containing a diverse set of query vocabularies, and find that QCEM-trained vectors had as good or better performance in almost all of the comparisons, and did particularly well on rare words, achieving a 4.8% increase.



(a) Learning with prior p_n^0

(b) Learning with optimized p_n

Figure 5.1: Toy example demonstrating the effect of negative sampling distributions on learning. Blue and red points are samples from p_d and the negative distribution. The green trajectory shows the optimization path of the model distribution's mean. a) Empirical selection of the sampling distribution results in a poor model fit. b) Optimized sampling distribution pushes away $p_m^{\boldsymbol{\theta}}$ more appropriately and results in a more accurate fit.

5.3. QUADRATICALLY CONSTRAINED ENTROPY MAXIMIZATION

We are given T i.i.d. data samples $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$ drawn from true but unknown data density $p_d(\boldsymbol{u})$ defined on the real domain $\boldsymbol{u}$. Similarly negative samples $\boldsymbol{y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T\}$ are drawn from the prior negative distribution $p_n^0(\boldsymbol{u})$. The goal is to fit a probability model $p_m^{\boldsymbol{\theta}}(\boldsymbol{u})$, having parameters $\boldsymbol{\theta}$. Without loss of generality of the framework, one can also learn unnormalized models which $\ln p_m^{\boldsymbol{\theta}}(\boldsymbol{u}) = \ln \tilde{p}_m^{\boldsymbol{\theta}}(\boldsymbol{u}) + \mathcal{Z}$ where $\tilde{p}_m^{\boldsymbol{\theta}}(\boldsymbol{u})$ represents the unnormalized density, and $\mathcal{Z}$ is the normalization factor to be learned. Then, the full parameter set to learn is $\{\boldsymbol{\theta}, \mathcal{Z}\}$. This leads to the negative sampling objective:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{p_d}\left[\ln \sigma(\boldsymbol{x};\boldsymbol{\theta})\right] + \mathbb{E}_{p_n^0}\left[\ln(1-\sigma(\boldsymbol{y};\boldsymbol{\theta}))\right] \tag{5.1}$$

Negative sampling is an instantiation of the contrastive framework. If we had unlimited data, for any sampling distribution, estimation error would be asymptotically normally distributed [94]. However, we are more interested in the word embedding problems where samples are usually considered to be insufficient for learning high-dimensional model densities. In such settings, our samples are finite, and biased.[1] If we have an unsuitable prior $p_n^0(\boldsymbol{u})$, the learned model $p_m^{\boldsymbol{\theta}}(\boldsymbol{u})$ can easily be inaccurate. For illustrative purposes, consider a toy scenario in Figure 5.1a where optimization is in the $\mathscr{R}^2$ space. Here, empirical samples obtained from p_d are highly biased and a naive negative sampling prior $p_n^0(\boldsymbol{u})$ is chosen for learning the model $p_m^{\boldsymbol{\theta}}(\boldsymbol{u})$. Negative sampling can not provide sufficient repulsion to stop $p_m^{\boldsymbol{\theta}}(\boldsymbol{u})$ from overfitting to the empirical samples. Instead, given a criterion to optimize the sampling distribution p_n, we could prevent the inaccurate model fits as in Figure 5.1b. This motivates one to optimize p_n before we perform stochastic updates to the embedding model.

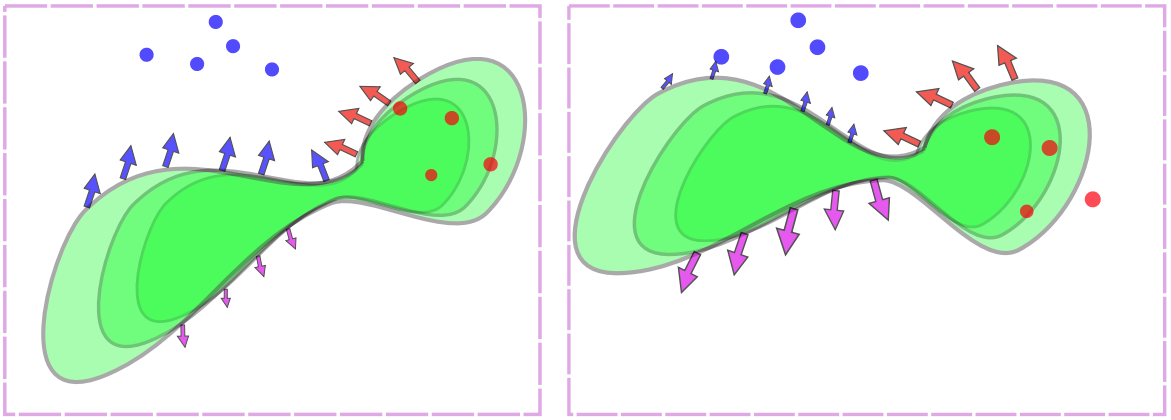
Although Equation 5.1 is the standard formulation of the negative sampling, we want to reformulate it to give us an intuitive understanding on the role of the negative distribution. To make the dependency on p_n explicit, we apply mechanical steps (provided

[1] Many cooccurrence statistics over word context pairs are either underestimated or overestimated.

in Supplementary Material) and rewrite Equation 5.1 jointly in terms of the embedding parameters $\boldsymbol{\theta}$ and the sampling distribution p_n:

$$\begin{aligned} J(\boldsymbol{\theta}, p_n) = \mathbb{E}_{p_d}[\ln p_m^{\boldsymbol{\theta}}(\boldsymbol{x})] - \mathbb{E}_{p_d}[\ln(p_m^{\boldsymbol{\theta}}(\boldsymbol{x}) + p_n(\boldsymbol{x}))] \\ - \mathbb{E}_{p_n^0(y)}[\ln(p_m^{\boldsymbol{\theta}}(\boldsymbol{y})) + p_n(\boldsymbol{y}))] + \mathbb{E}_{p_n^0(y)}[\ln p_n(\boldsymbol{y})] \end{aligned} \tag{5.2}$$

where we have four terms guiding the optimization of model distribution. With this reformulation, we can express the terms using a physical analogy, as attractive and repulsive forces. The first term is the fit term where we require the $p_m^{\boldsymbol{\theta}}(\boldsymbol{x})$ be similar to $p_d(\boldsymbol{x})$. In the second and third terms, the *mixture distribution* of $p_m^{\boldsymbol{\theta}}(\boldsymbol{u}) + p_n(\boldsymbol{u})$[2] is evaluated under the expectation of $p_d(\boldsymbol{u})$ and $p_n^0(\boldsymbol{u})$. This means, this mixture is repulsed to fit to these distributions and can be interpreted as terms to provide regularization to the learning of $p_m^{\boldsymbol{\theta}}$. We denote the second term as the *data repulsion* force, and the third term as the *prior repulsion* for the mixture distribution. If we analyze a single update on $\boldsymbol{\theta}$, model parameters, the fourth term becomes a constant. We can then illustrate in Figure 5.2 how the combination of three terms drives the optimization of the mixture distribution.



(a) Early Optimization (b) Late Optimization

Figure 5.2: Three forces guiding the optimization of mixture distribution $p_m^{\boldsymbol{\theta}}(\boldsymbol{x}) + p_n(\boldsymbol{x})$, shown in green contours. Blue and red are data and negative samples. Blue arrows represent the fit force, purple arrows represent data repulsion. Red arrows represent the prior repulsion. a) In early stages of optimization, fit force and prior repulsion push the mixture towards empirical samples. b) In later stages, data repulsion prevents overfitting to the data. Our goal here is to also optimize the data repulsion term to prevent overfitting to the data samples.

If we have not sampled any negatives from the prior $p_n^0(\boldsymbol{u})$, then the third and fourth terms do not contribute to Equation 5.2. In this scenario, the data repulsion term is the one preventing overfitting to the data samples. When we know that there is strong bias while sampling the data points, we have to learn p_n such that it provides sufficient data repulsion for the mixture $p_m^{\boldsymbol{\theta}}(\boldsymbol{u}) + p_n(\boldsymbol{u})$. This means we want to maximize the data repulsion $\mathbb{E}_{p_d}[\ln(p_m^{\boldsymbol{\theta}}(\boldsymbol{x}) + p_n(\boldsymbol{x}))]$ term for p_n. This is troublesome at first sight, since it looks difficult to disentangle the p_n function. Luckily, two design considerations in word embeddings allow us to bypass this problem.

[2]Mixture normalization constant is 2 but not shown for the ease of notation.

First, in many word embedding objectives, including Word2Vec [57] and GLoVe [93] embeddings, optimization is done on sufficiently high dimensional spaces, and model parameters are initialized randomly on the space [19]. Under this condition, we can assume that the model likelihood $p_m^{\boldsymbol{\theta}}(\boldsymbol{x})$ for any given sample will be negligibly low right after the initialization. Furthermore, p_n is usually constructed from the empirical distribution which means $p_n(\boldsymbol{x})$ is going to be the dominant term inside the mixture. These two common design practices allow us to instead optimize an upper bound. For any given data point $\boldsymbol{x}$, we consider $p_m^{\boldsymbol{\theta}}(\boldsymbol{x})$ as a constant and inferior quantity and write the upper bound as:

$$J(p_n) = -\mathbb{E}_{p_d}[\ln p_n(\boldsymbol{x})] \tag{5.3}$$

Since we are trying to maximize the objective, this equation suggests that we want to learn a p_n such that we want to deviate away from the empirical data distribution. The equation is underdetermined in nature; many choices are possible for selecting a negative sampling distribution p_n. We make the least possible assumptions, and resort to a Maximum Entropy [132] approach given that we satisfy distributional consistency. That is, we want to maximize the entropy of p_n, while being consistent with the empirical data's statistics. Optimizing the upper bound of the data repulsion term with respect to the empirical statistics, we aim to obtain a better sampling distribution for learning $p_m^{\boldsymbol{\theta}}$ as in Figure 5.1b. Reliance on the data's empirical moments will constrain the solution.

Assume the initial word frequencies are given in a data vector $\boldsymbol{d} = (d_1, d_2, ..., d_n)^T$ in which the entries are ordered: $d_i \geq d_{i+1}$ and where n is the vocabulary size. Let $\boldsymbol{p}$ be the parameters to be optimized for the p_n. We constrain the deviation of $\boldsymbol{p}$ from the data $\boldsymbol{d}$ by a quadratic constraint $(\boldsymbol{p}-\boldsymbol{d})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{p}-\boldsymbol{d}) \leq \beta n$ where $\boldsymbol{\Sigma}^{-1}$ is the precision matrix. These design considerations yield the following problem:

$$\begin{aligned} \max_{\boldsymbol{p}} \quad & \mathbb{H}[\boldsymbol{p}] \\ \text{s.t.} \quad & \boldsymbol{p} \geq 0 \\ & \mathbf{1}^T\boldsymbol{p} = 1 \\ & (\boldsymbol{p}-\boldsymbol{d})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{p}-\boldsymbol{d}) \leq \beta n \end{aligned} \tag{5.4}$$

where positivity and sum to one ensures that p_n is a probability function. Although this problem seeks sampling distributions with higher entropy, it is difficult to solve in practice via gradient descent updates. It frequently suffers from numerical difficulties when many probabilities are almost zero.[3] Then, the log function easily yields $-\infty$ values causing the gradient to go infinite where Lipschitz continuity conditions do not hold anymore. As the problem dimensionality increases, we are much more likely to encounter such problems. To circumvent problems arising from entropy maximization, we further want to design a surrogate for the problem in Equation 5.4.

Proposition 1. Let a probability mass function $\boldsymbol{p}$ defined with ordered probability masses: $p_1 \geq p_2 \geq ... \geq p_n > 0$. Then the application of a smoothing operator increases the entropy of $\boldsymbol{p}$.

[3] We know that word-context conditional distributions are highly sparse and contain very minor probabilities in their tail.

Proof The key part of the proof utilizes Taylor series expansion. The full proof is provided in Appendix B.2.

This result poses that there is a relation between the entropy and the smoothing operator. Motivated by it, we relax the entropy maximization problem in Equation 5.4 to:

$$\begin{aligned} \max_{\boldsymbol{p}} \quad & -\left\|(\boldsymbol{\Omega}-\boldsymbol{I})\boldsymbol{p}\right\|_2 \\ \text{s.t.} \quad & \boldsymbol{p} \geq 0 \\ & \mathbf{1}^T\boldsymbol{p} = 1 \\ & (\boldsymbol{p}-\boldsymbol{d})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{p}-\boldsymbol{d}) \leq \beta n \end{aligned} \tag{5.5}$$

where $\boldsymbol{\Omega}$ is chosen as a Hankel matrix [133]. This formulation enforces that neighboring entries in $\boldsymbol{p}$ become similar, making the distribution smooth and thereby increasing the entropy. Moreover, the problem is convex in $\boldsymbol{p}$ and known to yield a unique maximum [134]. This formulation does not make any distributional assumption on the form of p_n, nevertheless we can still favour particular solutions by setting the precision matrix $\boldsymbol{\Sigma}$. Using a Hankel matrix in its most general form results in an impractical number of objective terms for large vocabularies. Thus, we further embed a binary structure with $\boldsymbol{\Omega}_{ij} = 1$ if $j = i+1$ and $\boldsymbol{\Omega}_{ij} = 0$ elsewhere[4]. This specialized circulant structure of $\boldsymbol{\Omega}$ reduces the number of terms in the objective to n, the vocabulary size.

Proposition 2. Let a PMF $\boldsymbol{p}$ given with ordered masses: $p_1 \geq p_2 \geq \ldots \geq p_n > 0$. Also let $0 < \lambda < 1$ be the density powering parameter. Then, application of powering acts as a smoother on the density given that there exists a lower bound γ on p_i that it is related to λ with: $\gamma = \left(\frac{1}{\lambda}\sum_j p_j^{\lambda}\right)^{1/(\lambda-1)}$

Proof The proof follows by recognizing the Lipschitz condition, enforcing it to hold by assuming a lower bound and exploiting the diminishing structure of the first order derivative. The full proof is provided in Appendix B.3.

This result sheds light on why the heuristics [57, 93] adopted for the negative sampling works moderately well in practice. As long as the minimum probability mass of the sampling distribution is bounded, powering distributions acts as a smoother. This is simply an approximation to our smoothing formulation.

Despite its practical consequences, the problem with powering heuristic is that, to the best of our knowledge, there is no rationale for optimal sampling distribution to be in the Pareto family. Unlike [57], which constrains the word frequency density to be in the Pareto family, the formulation in Equation 5.5 yields more generality. It does not enforce any distributional assumptions, opening up possibilities to discover better optima. In the next section, we experimentally compare these heuristic approaches to our formulation.

[4] As Hankel and Toeplitz matrices are closely related, one can question the effect of $\boldsymbol{\Omega}$ being Toeplitz when using this binary structure. In this case, we achieve the same objective with Equation 5.5 given that entries of $\boldsymbol{p}$ are reversed. Hence for penalizing the difference of consecutive entries, choosing between Hankel and Toeplitz matrices does not constitute a key difference in our formulation, and is a matter of reparametrization.

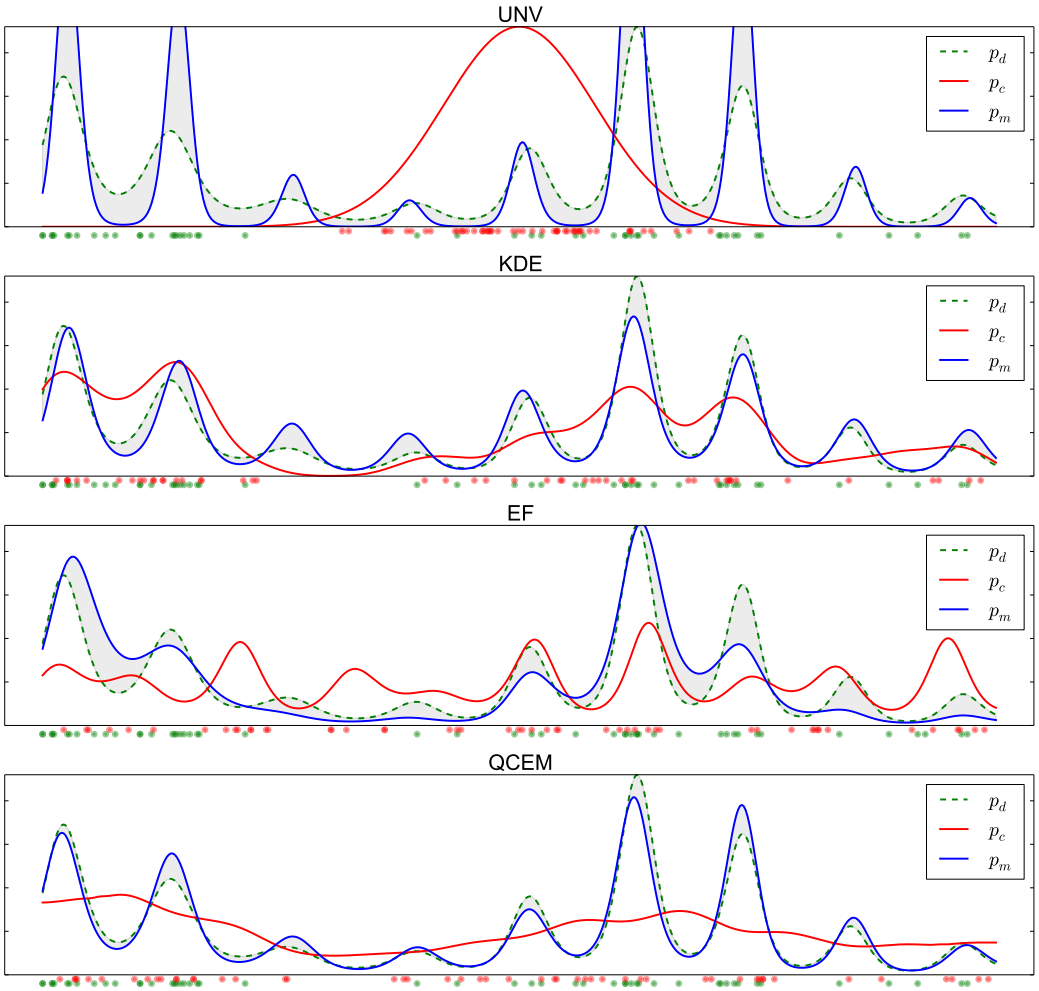


Figure 5.3: Learned models (blue) for the data density (green dashed), using different sampling distributions p_c (red). Top to bottom row shows a) Univariate b) KDE c) EF d) QCEM distributions. Green points are data samples $\boldsymbol{x}$ and red points are negative samples from p_c. Gray areas highlight the fitting errors.

5.4. EXPERIMENTS

Experimental Setup. We provide two sets of experiments to demonstrate the efficiency of our approach. For both experiments, the QCEM formulation is solved by a Splitting Conic Solver [135] that can solve large-scale convex cone programs by using an alternating directions method [136]. For synthetic experiments, we use inverse transform sampling to sample from 1D probability distributions. The error for model fits is measured by calculating the average $KL(p_d||p_m^{\boldsymbol{\theta}})$ by repeating the experiments 10 times with different random initializations. Learning the model distribution on both synthetic and real world experiments, we use the same stochastic gradient algorithm with the same learning rate for all settings.

5.4.1. EXPONENTIAL FAMILY DENSITY ESTIMATION

Data Generation and Parameters. The interest in this subsection is to quantify the contribution of QCEM contrastivity for the unnormalized density estimation problem. We define a data generator signal $S(\boldsymbol{\theta}^*, \boldsymbol{\phi}(u))$ over the domain $[-2\pi, 2\pi]$ with sine and cosine

bases:

$$S(\boldsymbol{\theta}^*, \boldsymbol{\phi}(u)) = \theta_1^* \sin(2\pi\omega_1 u) + \theta_2^* \cos(2\pi\omega_2 u) + ... + \theta_{2n}^* \cos(2\pi\omega_{2n} u)$$

where $\boldsymbol{\phi}(.)$ represents the transformation to the trigonometric functions. Then the probability densities are constructed using the Exponential Family (EF) representation:

$$p_d(u;\boldsymbol{\theta}^*) \sim \exp\left(S(\boldsymbol{\theta}^*, \phi(u))\right)$$

where trigonometric bases are interpreted as sufficient statistics. Finally, we learn the unnormalized EF density $\ln p_m^{\boldsymbol{\theta}}(u) = \ln \bar{p}_m(u;\boldsymbol{\theta}) + \mathcal{Z}$ with parameters $\{\boldsymbol{\theta}, \mathcal{Z}\}$. In other words, the goal is to learn the true canonical parameters of p_d, the amplitudes of each trigonometric statistics, plus the normalization constant of the density.

Methods. Our first baseline for the contrastive density is the univariate Gaussian density (UNV). Although it is simple to draw samples from this distribution, it is a poor choice for a contrastive function because it is only able to provide a limited amount of discrimination between data and contrastive densities. Another baseline choice of p_c is a more flexible nonparametric kernel density estimate [137] (KDE), where p_c is fitted to the observations. In some applications, one might know the parametric family of the underlying data density in advance, but not its parameters. We depict this case with an Exponential Family (EF) baseline where we have access to the true sufficient statistics of p_d, but not the canonicals. Knowing the true sufficient statistics of p_d is a very strong assumption, making this baseline very competitive. As the synthetic experiments have relatively low numerical complexity, we also report baseline results for the Maximum-Entropy (ENT) baseline (solution of the Equation 5.4). Finally, QCEM corresponds to our approach with an isotropic precision. We constructed data constraint vector $\boldsymbol{d}$ for this problem using the Kernel Density Estimate.

Results. Figure 5.3 shows the density fits obtained with each negative sampling approach. We observe that the univariate approach can only learn the prominent peaks of p_d in locations with many samples. For instance, the data peaks on the leftmost region are not captured accurately. In contrast, EF collects samples more homogeneously with its trigonometric bases and helps to fit more accurate models compared to the Univariate approach. KDE also obtains a fit that is comparable with the EF and QCEM fits. Using KDE, the low probability region variations are captured, but the probabilities of data peaks are not correctly estimated. QCEM contrastivity obtains the best fits: not only the data peaks, but also the low probability regions are captured much more accurately compared to the KDE and EF fits.

Note that the distribution p_c obtained by QCEM is relatively uniform compared to the EF and KDE distributions. This might raise the argument that a naive uniform distribution would provide the best sampling. Indeed, without any imposed moment constraints on the optimized distribution, the maximum entropy distribution is the uniform distribution. In a low dimensional setting the uniform distribution is an appropriate choice, but in high dimensions it quickly becomes problematic. Uniform sampling from a high dimensional volume is very inefficient, and a huge number of samples is required to ensure that we sample from regions where the data probability is sufficiently high. In contrast, QCEM combines the efficient sampling of data while providing homogeneous cover for the probability domain.

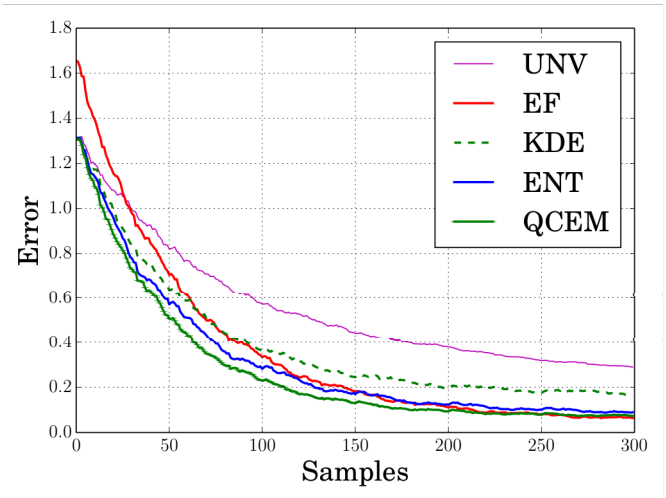


Figure 5.4: Learning curves of each contrastivity approach.

The full learning curves of all methods are depicted in Figure 5.4. Consistent with the findings of [138], asymptotically, all approaches are able to find the underlying density. Nevertheless, Univariate and KDE convergence is much slower than the other methods and they are inappropriate sampling techniques for small datasets. The EF and ENT approaches have a moderate rate of convergence. Note that the ENT approach has slower convergence, presumably due to the numerical difficulties of entropy maximization [139]. QCEM objective avoids these numerical problems, yielding a faster alternative to these approaches.

5.4.2. WORD EMBEDDINGS SIMILARITY

Data and Parameters. In the word embedding problem, the joint density over the sampled words and context pairs have to be learned. Following the state-of-the-art embedding evaluation schemas [96], we apply standard HTML text processing to Wikipedia. We remove words that occur less than 100 times in the whole corpus. This results in a sequence of several billion words, with a vocabulary size around 37k. The cooccurrence is then computed using windows of 10 tokens to each side of the focus word, following the practices of [140]. We use the word embedding architecture [59] that is known to be more robust for small sample sizes, dropouts and perturbations in the training set. Learning rate is initially set similarly to the methods and decayed in a linear fashion.

Evaluation and Baselines. Despite the challenging nature of the objective evaluation of learning the word vectors, recent work in [96] suggests that intrinsic tasks, such as word similarity measurements, are a better proxy for measuring the generic quality of word vectors than the extrinsic evaluations. We therefore follow the experimental setup of [96, 97], and compare the Spearman's correlation estimates of each model to human estimated similarities. Here a higher score indicates a higher correlation to human estimated word similarity judgements. For datasets containing multiple human annotators, we simply average the annotator scores.

The WSS and WSR [97] are similarity and relatedness subsets of WordSim353 [123] dataset. WSS contains taxonomic relations (e.g. synonymy) and WSR mainly covers top-

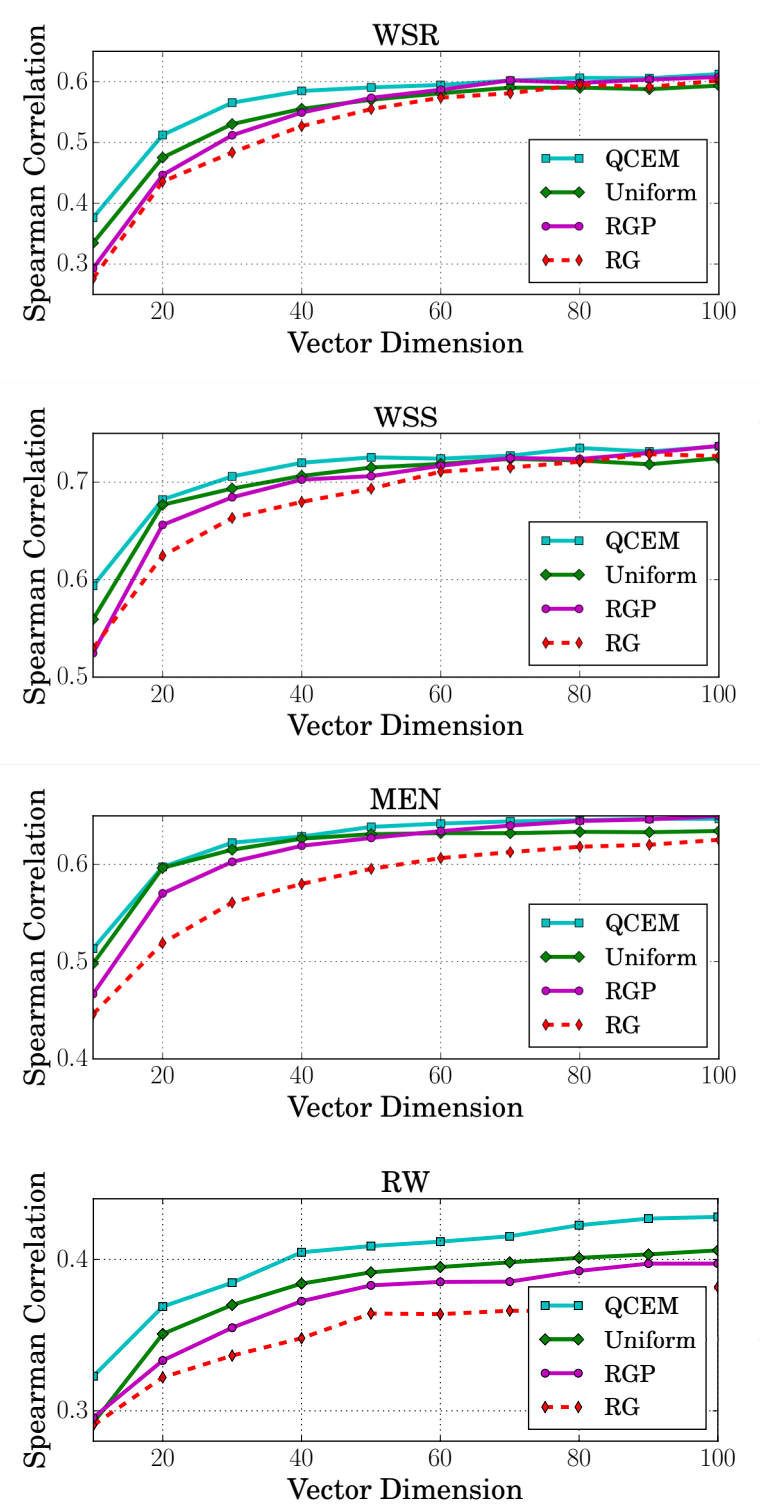


Figure 5.5: Word Similarity performances of the methods on WSR, WSS, MEN and RW datasets.

ical relations. These two datasets are relatively small and contain words that have relatively high frequency. MEN [84] word pair dataset contains 3k randomly sampled words, that occur at least 700 times, extracted from a freely available combined corpora having approximately 2.7B tokens. Sampling was performed to ensure balanced range of relatedness levels. The human similarity scores for this dataset are annotated using an interface for the Amazon Mechanical Turk. RW [141] dataset contains 2034 word pairs, first word randomly sampled from Wikipedia documents. Then the outliers are filtered using WordNet entries, and second word is sampled from synonym sets. Both MEN and RW contains many words with low frequency.

Baselines. We compare the following methods:

- *RG*, which uses the word frequency distribution, the data statistics, as negative distribution p_n.
- *RGP*, uses a power heuristic of the unigram distribution. The powered version of the word frequencies are used $\sim p_c(\boldsymbol{w})^{\lambda}$. This heuristic is the common baseline that is used by the state-of-the-art method [57], where λ is a corpus dependent parameter. For a fair comparison, we set λ accordingly to the empirical findings of [57, 93] as it is known to yield the best results for English corpora.
- *Uni* (Uniform) approach. We use an uniform distribution which all words of the vocabulary are equally probable to be picked as contrastive samples.
- *QCEM*, our proposed approach. For the problem construction, we use the unigram frequencies as data constraints: $\boldsymbol{d} \sim \sum_r \boldsymbol{C}_r$ where $\boldsymbol{C}_r$ are rows of word cooccurence matrix. For scalability considerations, we further speed up the approach by considering the optimization over equivalence classes over words. These equivalence classes are defined such that words having the same corpus frequency are treated as the same class. This equivalence strategy yields 5.2k variables to optimize instead of the 37k variables, adding an order of magnitude speed increase. Finally, we did not assume any apriori precision and decided to use an isotropic $\boldsymbol{\Sigma}$.

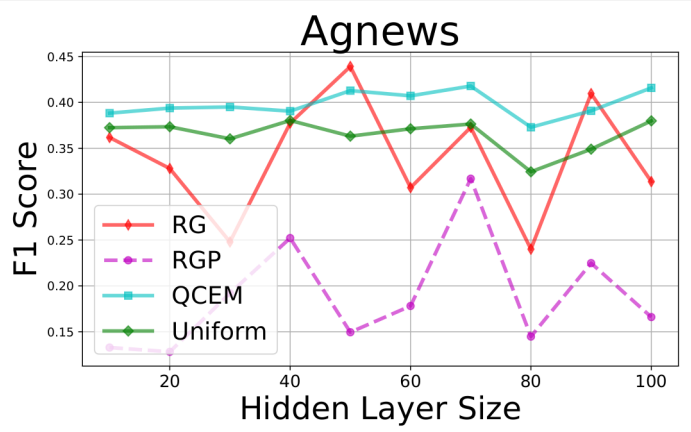
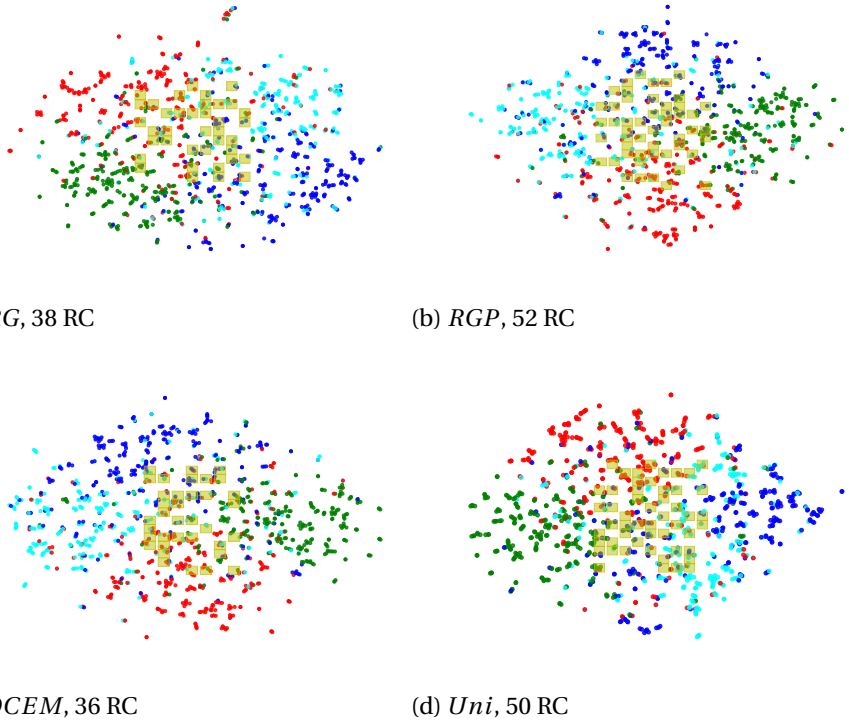


Figure 5.6: Agnews text classification performance of vectors trained with each sampling algorithm.

Quantitative Results. Figure 5.5 shows the word similarity performances for all approaches on all datasets. In all datasets, the *RG* baseline performs poorly. For simpler



(a) *RG*, 38 RC (b) *RGP*, 52 RC

(c) *QCEM*, 36 RC (d) *Uni*, 50 RC

Figure 5.7: T-SNE dimensionality reduction of document embeddings. Each class is coded with a color. As confusions are common in the center region, we quantify the number of confusion regions (samples from multiple classes are present). Yellow boxes indicate Region of Confusions (RC). Less clutter is observed for *QCEM* embeddings.

datasets such as WSR and WSS, *QCEM* outperforms all baselines, especially on the lower dimensional regime where the correlation gain is slightly larger than the high dimensional regime. Both taxonomy relations in WSS, and topical relations of WSR gain from the *QCEM* sampling. The *Uni* approach yields competitive performance especially in lower dimensions, but on high dimensional data the performance degrades quickly, as in the WSR and MEN datasets.

The performance gaps become more perceivable on more difficult datasets. On the MEN dataset, *RGP* is worse than *Uni* especially in lower dimensions, whereas in high dimensions the powering approach is better than the uniform distribution. *QCEM* and *Uni* perform quite alike in low dimensional MEN experiments. We believe this is due to two reasons. First, MEN similarity scores are much more noisy than other WS datasets due to the non-expert annotators which conceals the performance gap. Secondly, MEN vocabulary content is much broader than other datasets and it contains words occurring more than 700 times. This means query words are mostly from the heavy tail region in which *QCEM* and *Uni* behaves similarly. Nevertheless, *QCEM* does not suffer from performance losses in high dimensions like *Uni* approach and consistently achieves better performance.

On the RW dataset, it is noteworthy that the uniform contrast approach outperforms the powering heuristic with a small margin, for all model instances. For the WSS and WSR datasets, the powering heuristic obtains a reasonable performance whereas in the

RW dataset it performs worse. Apparently, the constraints imposed by the powering heuristic turns out to be inappropriate for the RW dataset. This results in a suboptimal solution when the semantic relations of words are queried for a large set of less frequent words. The *QCEM* approach, on the other hand, does not impose such constraints, and obtains performance improvements with large margins. In the RW dataset, we finally compute the average correlation score over all the models, resulting in a 2.0% increase over the powering heuristic and a 4.8% over the standard baseline, a powerful quantitative indicator that embeddings trained with *QCEM* yields more realistic structure than the ones trained with computationally simple, but theoretically not justified heuristics.

5.4.3. REAL WORLD TEXT CLASSIFICATION

Setup. We evaluate vectors in Agnews text classification benchmark which consists of news articles collected from multiple sources. The dataset is randomly split into 120k training and 7k test documents and the goal is to predict the label of each document from {world,sports,business,science-technology} classes.

We plug in trained word vectors to a standard Multi Layer Perceptron (MLP) with logistic activation units and ensure fair comparison by fixing the embedding weights during the training which means word vector layer does not change. This helps us to accurately quantify the performance gain from input vectors. Experiments are carried with varying number of hidden units to evaluate how vectors contribute to different type of networks and whether provide sufficient generalization for different architectures. Each network is then trained using a standard Stochastic Gradient Descent optimizer with adaptive learning schema. We then compute the F-1 scores for each approach.

Results. The result of each experiment is shown in Figure 5.6. *RGP* approach performs worse in general. Networks trained with *RG* vectors occasionally perform well, but perform poorly on average. These vectors suffer from performance fluctuations suggesting that they are less robust to the local minima inherent in the problem. We also observe this phenomenon when we use vectors with *RGP*, illustrating another reason why optimizing the sampling distribution with our approach is advantageous. Note that performance of *QCEM* does not deteriorate even for networks with large number of neurons, and produces more stable scores. We visualize the documents constructed from embeddings. In Figure 5.7, we show dimensionality-reduced document vectors in which each yellow region denotes a Region of Confusion. We expect document embeddings to have low intra-class distances, and high inter-class distances. *QCEM* document clusters are more coherent, and subject to less confusion in the center region.

5.5. CONCLUSIONS

We have presented a novel framework for optimizing negative sampling distribution using our Quadratically Constrained Entropy Maximization (QCEM) approach. Our formulation posed a convex and computationally attractable solution, has linear time complexity with respect to the vocabulary size, and permits scaling to large word embedding problems. Our theoretical analysis showed not only the generality but also the relation of our work to the prior heuristic state of the art approach, that is shown to be an approximation to our general maximization framework.

We validated our formulation both in synthetic density and real-world word vector space learning experiments, demonstrating that QCEM obtains faster convergence rates compared to a various competing approaches for learning exponential family probability densities. Finally we reported the performance of QCEM in word similarity tasks, in which the restrictive probabilistic assumptions of the heuristic methods were not fulfilled whereas our approach with its generality performed significantly competitive than the heuristics and entropy promoting baselines. Combination of the theoretical results and empirical evidences obtained for the vector space learning problems suggests that QCEM is an attractive solution to apply for determining negative sampling distributions.

6

CONSTRAIN GLOBAL SAMPLE LOCAL: FASTER AND VARIANCE REDUCED WORD EMBEDDINGS

6.1. ABSTRACT

Word embeddings represent the words in a language with feature vectors. They are typically learned with sampling approaches that approximate the partition function of a high dimensional probabilistic model. Nevertheless, in this chapter we first show that ordinate sampling approaches are not guaranteed to produce robust embeddings. First we show that in the low sampling budget regime, sampling instances produce inaccurate approximations and does not necessarily follow the global behaviour of the infinite sampling. Second, samplers are mostly designed to be context-free which does not exploit local relevancies in the embedding space. We first address these limitations by introducing target variables which are used for imposing global constraints on low budget sampling. This is followed by incorporating local contextual relevance to our sampling distribution by grounding on word concreteness knowledge bases. Our experimental results show that embeddings produced by *Constrain Global Sample Local* (CGSL) is highly competitive to the baseline sampling approaches on a suite of intrinsic word similarity tasks. Furthermore, a quantitative analysis on CGSL reveals not only its potential to produce robust embeddings due to its variance reduction effect but also its apt to converge faster to the gold standard word vectors.

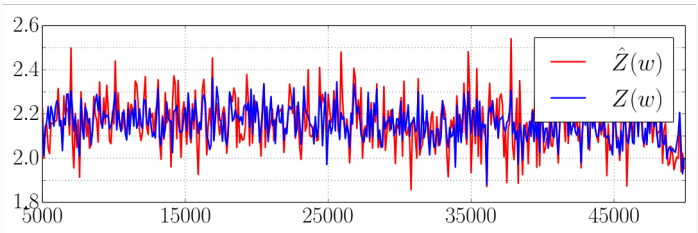


Figure 6.1: Exact $Z(\boldsymbol{w})$ and approximated $\hat{Z}(\boldsymbol{w})$ partition function for all words in the vocabulary. $\hat{Z}(\boldsymbol{w})$ resulting from the finite sampling budget frequently overshoots the $Z(\boldsymbol{w})$ for many words.

6.2. INTRODUCTION

Word embeddings represent the words in a language with vectors learned from the cooccurence statistics of a corpora. Statistical approximations enable learning of these vectors from large text collections containing billions of tokens. These vectors are then conveniently used as low level features for a set of downstream natural language processing tasks; such as dialogue systems [142], question answering [128], sarcasm detection [85], and discourse relation recognition [143].

A large variety of word embedding methods are based on the exponential family word vector space model equipped with a partition function (PF). Unfortunately this PF is almost always intractable in practice, due to the large number of words in the language. This motivates sampling as the common statistical approximation for the word embedding models. Sampling routines approximate the PF by drawing few negative samples from a user specified sampling distribution. Although these routines provide practical and efficient approximations, to the best of our knowledge, the full implications of the word vector sampling is not understood to the full extend [144].

In this chapter, we first investigate the limitations of typical sampling under small sampling budgets, and demonstrate that naive sampling applications does not produce robust approximations for the PF. We show that for a large variety of word context occurrences, unbounded approximations of these sampling instances cause significant deviations from the true value of the PF.

In Figure 6.1, we depict the usual scenario without any precautionary mechanism. Deviations are prone to trigger overshoots in the approximation and introduce uncertainty to the training that is detrimental for learning robust word vectors. Furthermore, another prominent limitation of current sampling methodologies is their context-free nature. They usually do not exploit local relevancies of words in the embedding space. We hypothesize that insufficient discrimination arising from such context-free comparisons result in ineffective usages of the budget, harming the learning process.

We develop a novel sampling methodology that simultaneously addresses both limitations of the word vector sampling. Our technique first constructs global target variables acting as steady points during the optimization. Grounding on these variables, we globally constrain the sampling sequences, and eliminate inordinate ones that is accountable for large gaps in the approximation. Given these reliable sequences, the second aforementioned limitation is addressed by first discovering that the word concreteness provides a weak form of local contextual relevance in the word embedding problem. We formulate and construct a sentence concreteness model, which is injected to

the process by adaptively updating the sampling distributions. Unlike context-free sampling, our sampling distributions considers local contextual relevance while accounting for practical efficiency.

The potential merits of *Constrain Global Sample Local* (CGSL) are empirically validated on a large suite of word similarity tasks. Under varying amount of data, our embeddings are found to outperform the baselines on the dataset basis, and yield very competitive results on the dataset average. A further quantitative analysis on our sampler reveals how it effectively reduces the sampling variance which is an essential trait for obtaining robust embeddings. On top of that, we demonstrate locally relevant sampling leads to the faster convergence to the true, gold standard, embeddings.

6.3. CONSTRAIN GLOBAL SAMPLE LOCAL METHOD

In this section, we first introduce the sampling approximation gap problem, and then propose our novel sampling algorithm.

6.3.1. SAMPLING APPROXIMATION GAP

Let V be the vocabulary size of a dataset, d the embedding space dimensionality and N the number of training samples. Let $\boldsymbol{w}, \boldsymbol{c}$ be discrete word indicators for the context and target vectors and, similarly, let $\boldsymbol{\Phi}, \boldsymbol{\Psi}$ be the context and target embedding matrices that contain the embedded vectors $\vec{\boldsymbol{w}}$ and $\vec{\boldsymbol{c}}$. K denotes the number of negative samples, or the *sampling budget* for the sampling distribution. We consider the exponential family word vector space model [13]:

$$p(\boldsymbol{w}|\boldsymbol{c}) = \frac{\exp(\vec{\boldsymbol{w}} \bullet \vec{\boldsymbol{c}})}{\sum_{\boldsymbol{c}_n \in V} \exp(\vec{\boldsymbol{w}} \bullet \vec{\boldsymbol{c}}_n)} \tag{6.1}$$

where the conditional probability of observing a word $\boldsymbol{w}$ given a context embedding $\boldsymbol{c}$ is an exponential map. For convenience, we call the function $\exp(\vec{\boldsymbol{w}} \bullet \vec{\boldsymbol{c}}_n) = Z(\boldsymbol{w}|\boldsymbol{c}_n)$. The denominator term $\sum_{\boldsymbol{c}_n \in V} Z(\boldsymbol{w}|\boldsymbol{c}_n) = Z(\boldsymbol{w})$ is referred as the partition function, a summary statistics for the exponential map taking all possible contexts into account. In particular, note that $Z(\boldsymbol{w})$ has to be computed for all words $\boldsymbol{w}$ which is computationally very intensive. In order to approximate $Z(\boldsymbol{w})$, a negative sampling distribution $p_n(.)$ is specified [57]. The approximation takes the following form:

$$\hat{Z}(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{c}_n \sim p_n}[Z(\boldsymbol{w}|\boldsymbol{c}_n)] \tag{6.2}$$

In practice, the expectation in Eq. 6.2 cannot be computed in closed form for a large set of the p_n distributions. Rather, a sampling budget of K is maintained and few context words are drawn from the sampling distribution $p_n(\boldsymbol{c}_n)$ to yield an empirical estimate:

$$\hat{Z}(\boldsymbol{w}) = \frac{1}{K} \sum_{n}^{K} Z(\boldsymbol{w}|\boldsymbol{c}_n). \tag{6.3}$$

Although it is feasible to approximate $Z(\boldsymbol{w})$ with this quantity, there is no convincing reason for these approximations to be accurate, and it is questionable whether it does sufficiently well for many words in the vocabulary. In Figure 6.1, we demonstrate $\hat{Z}(\boldsymbol{w})$

for all w words in the vocabulary. Observe that the approximation easily undershoots or overshoots the true $Z(\boldsymbol{w})$, it is admissible to claim that overshoots get even more critical as the sampling budget gets significantly low. We refer to this gap as the *approximation gap*, a quantity that needs to be maintained in order to guarantee an accurate estimate of the partition function.

6.3.2. GLOBAL BANDS FOR APPROXIMATION GAP

To maintain the approximation gap, we are required to exploit the structure of the partition function. Empirically, we verify that in many contexts, $Z(\boldsymbol{w}|\boldsymbol{c}_n)$ concentrate around particular values [145]. The distribution for the random variable $Z(\boldsymbol{w}|\boldsymbol{c}_n)$ yields a Gaussian distribution with a mean $Z(\boldsymbol{w})$ and a finite variance. As the variance for all words are bounded, one can safely target for shrinking this gap during the sampling step. However, since we are prohibited to access $Z(\boldsymbol{w})$ as the target variable directly due to the computational reasons, we instead introduce a global target variable, that is a surrogate quantity to it:

$$\tilde{Z}(\boldsymbol{w}) = \frac{1}{M}\sum_{n}^{M} Z(\boldsymbol{w}|\boldsymbol{c}_n) \tag{6.4}$$

with the condition that $M \gg K$. This is a more reliable estimate of $Z(\boldsymbol{w})$ with a boosted sampling budget where M is a free parameter that can be arbitrarily tuned to increase the reliability of the estimate.

The reader here must note that using the intermediate statistic $\tilde{Z}(\boldsymbol{w})$ is cheaper from doing naive sampling with a budget of M. Thus, the contribution here is due to the fact that $\tilde{Z}(\boldsymbol{w})$ is only updated globally, while $\hat{Z}(\boldsymbol{w})$ is locally updated *per training sample*. Computing $\tilde{Z}(\boldsymbol{w})$ is not a bottleneck for our approach due to this lazy global update.

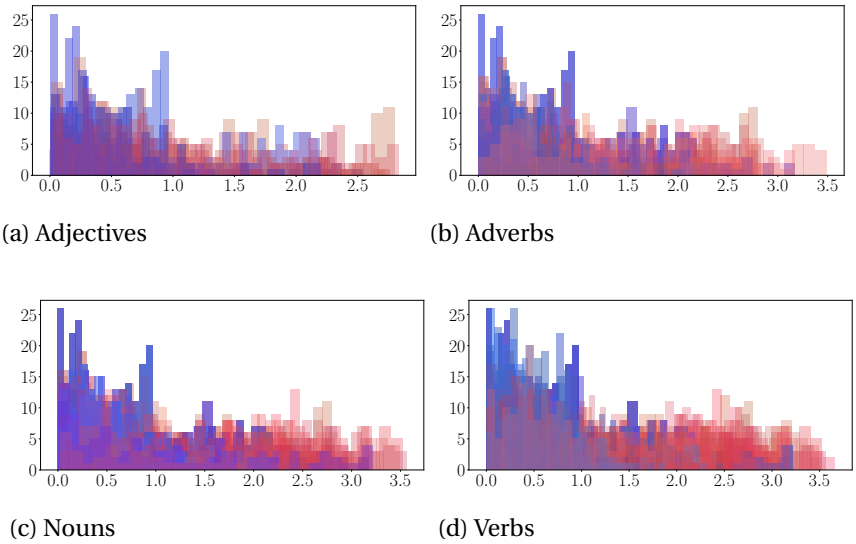
In order to see how the lazy surrogate $\tilde{Z}(\boldsymbol{w})$ in Eq. 6.4 is a more reliable estimate for $Z(\boldsymbol{w})$, we can resort to the standard argumentation of the Law of Large Numbers [146] stating the following inequality:

$$|Z(\boldsymbol{w}) - \tilde{Z}(\boldsymbol{w})| \leq |Z(\boldsymbol{w}) - \hat{Z}(\boldsymbol{w})|$$

which holds with high probability. Armed with this sampling argument, we can then use $\tilde{Z}(\boldsymbol{w})$ and instead perform guidances to the sampling to target for closing the gap $|\tilde{Z}(\boldsymbol{w}) - \hat{Z}(\boldsymbol{w})|$ using our target variables. There are two key benefits with this strategy. First, we are still able to use the same K-budget sampling, on sentence basis, to generate negative samples with a smaller gap, and thus avoid the full computation of $Z(\boldsymbol{w})$. Secondly, and more importantly, we can interpret shrinking the gap $|\tilde{Z}(\boldsymbol{w}) - \hat{Z}(\boldsymbol{w})|$ as doing variance reduction [147] which variance of the estimator is $\mathbb{E}\left[(\tilde{Z}(\boldsymbol{w}) - \hat{Z}(\boldsymbol{w}))^2\right]$ for the negative sampling. This feature is expected to give more reliable results during optimizing the word vectors.

6.3.3. LOCAL CONTEXT RELEVANCE VIA CONCRETENESS

In the previous subsection, we addressed the approximation gap by introducing global target variables and aiming to shrink the estimation gap. This methodology prevents large overshoots in the sampling by eliminating inordinate sequences. Since a global constraint over the word space is imposed, we reassure an amount of global consistency



(a) Adjectives (b) Adverbs

(c) Nouns (d) Verbs

Figure 6.2: Histograms for the concreteness difference of a word with its local neighbours (shown with blue tones) and random words (shown with red tones). Observe that densities are separable, and word concreteness provides strong contextual relevance. The relevance is especially greater for nouns and verbs.

during the sampling. Nevertheless most of the natural languages are characterized by a vast number of words, and the sampling distributions constructed from the vocabulary has a heavy tail. In this circumstance, random sampling disregards the contextual dependencies between words. In a context free setting with a large number of words, there are vast possibilities for sampling a negative word but no binding factors for obtaining sufficient discrimination. In [148], similar difficulties were discussed and the problem is identified as the *zero gradient problem*, suggesting that contextually inappropriate updates are uninformative for learning and waste the sampling budget in upcoming training iterations.

We hypothesize a more appropriate sampling should introduce a form of contextual dependency by exploiting local word relevancies, to ensure sufficient discrimination. We want to quantify the contextual relevance of a negative sample in order to avoid aforementioned problems and incorporate the context back to the sampler. For this reason, we utilize the concept of word concreteness, which is defined as *the degree to which the concept denoted by a word refers to a perceptible entity* [149]. However, we need to check whether we can associate word concreteness to the local relevance.

In Figure 6.2, we downloaded well-trained large scale GoogleNews word vectors [150], interpret the word neighbourhood in high dimensional embedding space as its local context, and measure the difference of concreteness score [151] the word and its neighbouring words. We repeat the analysis for hundreds of random words having four standard part-of-speech tags, and plot the histogram of concreteness differences. Histograms obtained for local neighbourhood are drawn in blue tones, and histograms obtained from a random neighbourhood are drawn in red tones. Observe that the concreteness difference is much lower for blueish histograms, especially for nouns and verbs in which the large differences are observed for the context-free sampler.

Since concreteness of words provides us clues on their local relevance in our framework, the inefficient sampling pairs are those that contrast abstract space words to concrete words. It is reasonable to claim that abstract entities such as *democracy* should be contrasted to the *freedom of speech* or *citizenship* rather than words of physical objects, and same holds for words referring to physical objects. By using concreteness scores, our globally constrained but also locally-aware sampler is expected to avoid such insufficient sampling pairs.

6.3.4. LOCALLY RELEVANT SAMPLING MODEL

We model the concreteness of each word with a Gaussian distribution and obtain the concreteness of a training sentence $S_{\boldsymbol{w}}$ using the following:

$$\Delta(S_{\boldsymbol{w}}) = \sum_{\boldsymbol{c} \in S_{\boldsymbol{w}}} KL\big(\mathcal{N}(\Delta_{\boldsymbol{w}}, \sigma_{\boldsymbol{w}}) || \mathcal{N}(\Delta_{\boldsymbol{c}}, \sigma_{\boldsymbol{c}})\big) \tag{6.5}$$

where $\{\Delta_{\boldsymbol{w}}, \sigma_{\boldsymbol{w}}\}$ are the concreteness mean and variance of the word $\boldsymbol{w}$. The sentence concreteness in Eq. 6.5 is then computed in closed form [152], and used for constructing the probability $p\big(\boldsymbol{c}_n | \Delta(S_{\boldsymbol{w}})\big)$. We then construct the novel sampling distribution which is factorized by this locally-aware density and context-free frequency distribution:

$$p_{\Delta}(\boldsymbol{c}_n) \sim p_n(\boldsymbol{c}_n)\, p\big(\boldsymbol{c}_n | \Delta(S_{\boldsymbol{w}})\big)$$

which generates samples locally relevant to the word of interest $\boldsymbol{w}$. Finally we are ready to provide the full form of our sampling algorithm. Remember that we had $\tilde{Z}(\boldsymbol{w})$ available for each word $\boldsymbol{w}$. Given few burn-in negative samples, our sampling addresses the gap $|\tilde{Z}(\boldsymbol{w}) - \hat{Z}(\boldsymbol{w})|$ in a time dependent manner. By τ_e we denote the $\tilde{Z}(\boldsymbol{w})$ which the sampling sequence should reach at the end of negative sampling process, and let τ_i denotes the initial sampling value $\hat{Z}(\boldsymbol{w})$. Given these initial and final references, we define a linear band over time:

$$\mathscr{L}(t) = \tau_i + t(\tau_e - \tau_i)/T$$

A sampling sequence is valid if it satisfies the enveloping condition:

$$|\tilde{Z}(\boldsymbol{w}) - \mathscr{L}(t)| < \omega$$

Here, the bandwidth ω of the linear function $\mathscr{L}(t)$ is directly determined from the sample variance of $\tilde{Z}(\boldsymbol{w})$, which we denote as $\bar{\mathbb{V}}[Z(\boldsymbol{w}|\boldsymbol{c}_n)]$. This is done for each time step t, and the sampling process terminates when the budget is filled. The pseudo-code of CGSL algorithm is provided in Algorithm 1.

Although it might be attractive to learn the context-dependent function, such as in [153], there are certain disadvantages of such a strategy. In fact, it is quite difficult to know and interpret what the sampling function learns especially when it has heavy tails. It is also not trivial to set the correct regularization parameters for learning architecture to enable efficient learning. Our strategy is to circumvent these approaches and stick to a sampling distribution which is more interpretable.

Computational Complexity. The usual bottleneck of sampling based statistical approximations is their computational complexity. We aim to retain the computational advantages of simple sampling techniques. Our CGSL sampler an instance of non i.i.d sampling, fast and achieves a time complexity of $O(VM + NK)$. The additional overload of

Algorithm 1: CGSL Sampling Algorithm

Input: $(\boldsymbol{w}, \boldsymbol{c}) \in X$: word-context pairs,
    K: sampling budget, $\mathscr{G}$: max epoch
Output: $\hat{\boldsymbol{\Psi}}$ word embedding matrix
Initialize matrix $\hat{\boldsymbol{\Psi}}$ with burn-in iteration.
repeat
  $\tau_e \leftarrow \tilde{Z}(\boldsymbol{w})$ using Eq. 6.4.
  $\omega \leftarrow \bar{\mathbb{V}}[Z(\boldsymbol{w}|\boldsymbol{c}_n)]$
  for all $(\boldsymbol{w}, \boldsymbol{c}) \in \mathscr{C}$ **do**
    Apply positive gradient of $(\boldsymbol{w}, \boldsymbol{c})$ to $\hat{\boldsymbol{\Psi}}$.
    Draw i.i.d budget $\boldsymbol{c}_n \sim p_n(\boldsymbol{c}_n)$.
    $\tau_i \leftarrow \hat{Z}(\boldsymbol{w})$ using Eq. 6.3.
    $\mathscr{L}(t) = \tau_i + t(\tau_e - \tau_i)/T$
    Compute $\Delta(S_{\boldsymbol{w}})$ using Eq. 6.5
    for all t **to** T **do**
      Draw $\boldsymbol{c}_n$ from $p_\Delta(\boldsymbol{c}_n)$ if $|\tilde{Z}(\boldsymbol{w}) - \mathscr{L}(t)| < \omega$.
      Update $\hat{Z}(\boldsymbol{w})$.
    end for
    Apply negative gradients $(\boldsymbol{w}, \boldsymbol{c}_n)$ to $\hat{\boldsymbol{\Psi}}$.
  end for
until $g > \mathscr{G}$ is $true$

$O(VM)$ is negligible due to the nature of word embedding problems, where the training set size is orders of magnitude larger than the vocabulary size in many languages [154]. Thus, the overall complexity is dominated by the term $O(NK)$. Consequently, our global update step that computes Eq. 6.4 does not pose a bottleneck and $O(VM+NK) \approx O(NK)$ holds.

6.4. EXPERIMENTS

Setup. We train our embedding models using the Wikipedia 2017 July snapshot. Our corpus preprocessing follows the standard state of the art practices for Wikipedia and the training parameters are set similarly as in [113]. For a fair comparison, same learning

Table 6.1: Mean and standard deviations of most, average and least concrete words of the dataset.

Word	MV Rating	Word	MV Rating
apple	5.0 ± 0	recompile	2.92 ± 1.23
boat	4.93 ± 0.37	surrogate	2.83 ± 1.28
milk	4.92 ± 0.39	sharpness	2.69 ± 1.34
side	3.68 ± 1.33	legalism	1.3 ± 0.76
symbol	3.11 ± 1.37	infinitive	1.27 ± 0.58
clean	3.07 ± 1.41	agnostically	1.19 ± 0.5

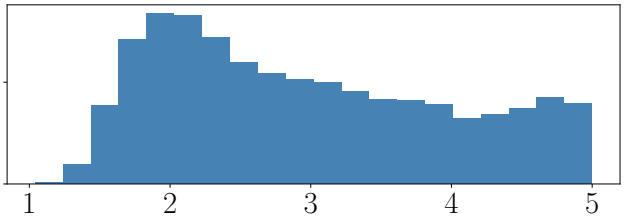


Figure 6.3: Distribution of word concreteness scores.

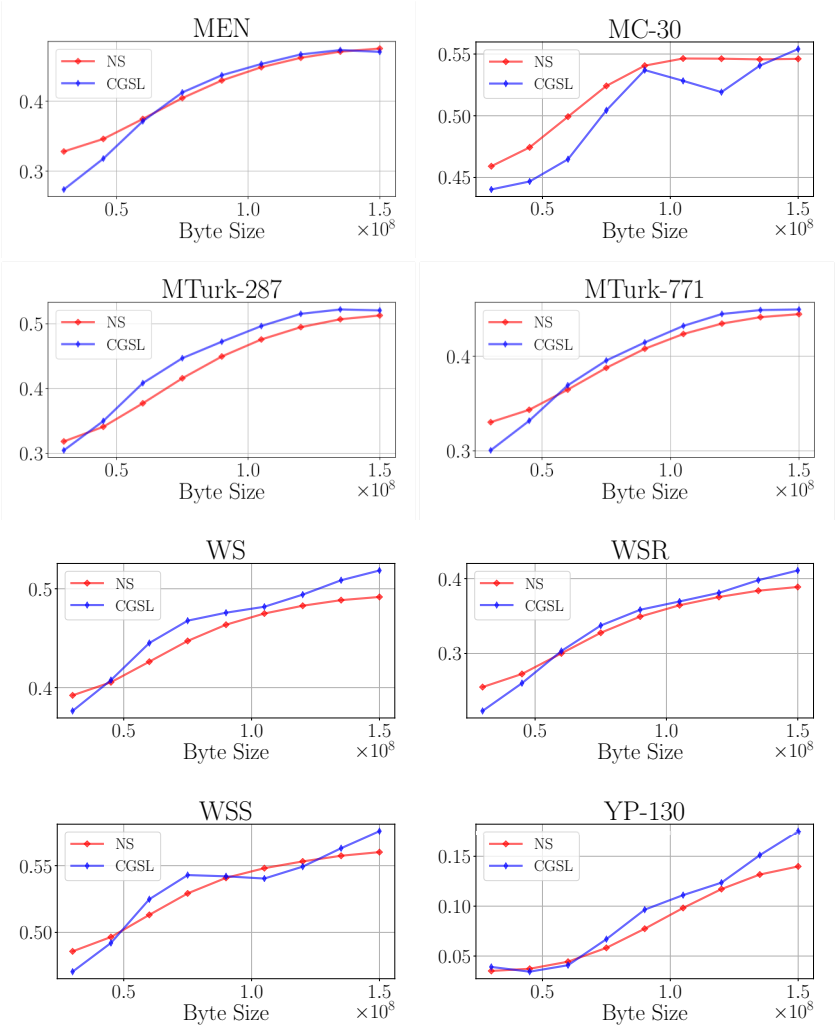


Figure 6.4: Word similarity performances of our approach on various datasets. We plot the learning curves; x-axis is the number of training bytes for the embeddings and y-axis is the correlation of embedding scores to the human judgements.

rates are applied for samplers with linear decays. Stochastic Gradient Descent is used for applying the gradient updates. We refer the negative sampling which acts as the state of the art baseline sampler as NS [99]. Also CG as the sampler which only uses Global Constraints, and finally CGSL as our approach.

We use the concreteness database [151] in which 40k English words have concreteness ratings from 1.0 (indicating fully abstract), to 5.0 (indicating fully concrete) collected from 30 human annotators with their standard deviations. In Figure 6.3, we illustrate the distribution of concreteness scores in the dataset. Table 6.1 shows examples of most, average and least concrete words of the dataset.

6.4.1. PERFORMANCE ON WORD SIMILARITY

As the off-the-shelf standard extrinsic word vector evaluation, we follow the standards of [96], but use a test suite by using the following datasets: MC30 [116], MEN [117], MT287 [118], MT771 [119], WordSim-353 [123]. We then vary the number of bytes of the training corpus, and measure the Spearman Correlation Coefficient after repeating the experiments for multiple times. The results are shown in Figure 6.4 and 6.6. For a collection of datasets such as MEN, MT771, WSR, WSS; we observe it takes sampler some amount of preliminary data to reach a satisfactory performance. This is to be expected since when the amount of training bytes is not significantly greater than the vocabulary size, it can be characterized as the unconfident regime for the non-standard samplers. As the amount of training data keeps increasing, embedding instances trained with CGSL approach outperforms its baselines with perceivable performance gains and on the dataset average. We also analysed bandwidth size's influence, where observations showed there is a linear relation between the number of rejected sampling sequences due to the global constraint and the specified bandwidth coefficient. Nevertheless, we found out that word similarity performances were insensitive to the specification of the bandwidth.

6.4.2. VARIANCE REDUCTION

After training the word vectors, we draw the same number of samples from the sampling distribution $p_n(\boldsymbol{c}_n)$ and calculate its $\hat{Z}(\boldsymbol{w})$ approximation for all words in the vocabulary. After repeating this experiment multiple times, we report the sampling variances in Figure 6.5. Observe that variances are much lower, especially for the infrequent rare words $\boldsymbol{w}$, for our case suggesting that addressing the approximation gap is more critical for these words.

In terms of the bias-variance tradeoff, our sampler reduces the variance of the estimate, at the cost of a small increase in bias. In fact, this is precisely what we aim for large text corpora learning as word embeddings samplers are already possesses bias to some extend [155]. Furthermore, we noticed that the word similarity results are not very sensitive to the specification of the M parameter. We plot this phenomenon in Figure 6.4 where $M = 0$ corresponds removing our global constraints. Since the performance is stable with varying values for M, the choice usually depends on the computational resources.

From an optimization point of view, variance reduction for a word embedding sampler is advantageous. When embeddings are optimized using online gradient techniques, learning rate specification is an empirically driven process that requires extensive ex-

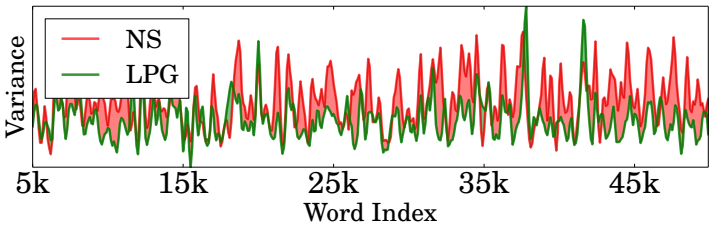


Figure 6.5: Sampler variances of NS and LPG.

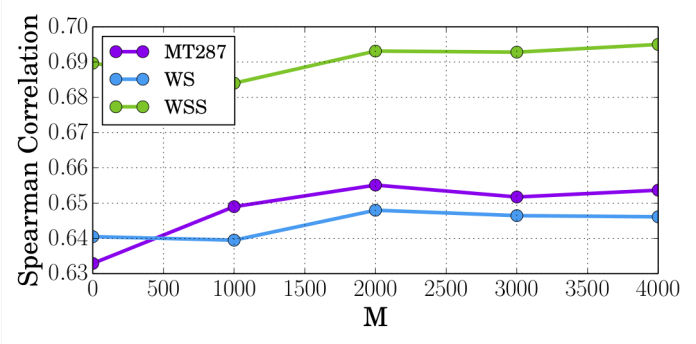


Figure 6.6: Effect of M to the performance.

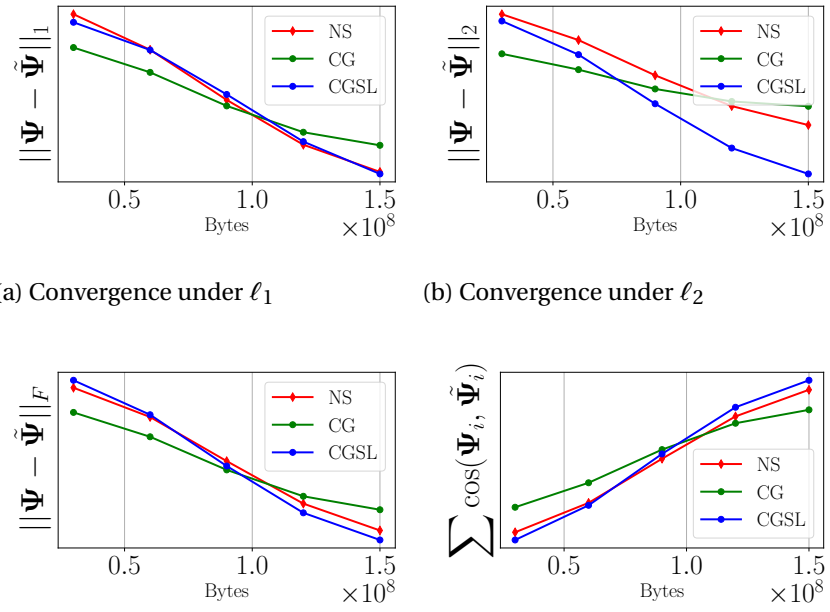
perimentation. Techniques such as dynamic learning rates on words [156], or adoption of memory based gradient optimizers [157] explicitly address this difficulty. On the other hand, variance reduction implicitly maintains it, and motivates training with larger learning rates, due to more reliable updates of the CGSL.

6.4.3. CONVERGENCE RATES

In subsection 6.4.1, we measured how our embeddings correlate to the human judgements. Another desideratum for embeddings is how fast they are learned. Here, we quantitatively measure the convergence rate by first training embeddings long epochs on the corpus and obtain embedding matrix $\mathbf{\Psi}$ that is assumed to be the gold standard, true embedding. Then we train embeddings but similarly to the settings in [158] apply early-stopping to measure which methods can reach the gold standard solution faster despite seeing the same amount of reduced training set.

We use the matrix discrepancy $\left\|\mathbf{\Psi} - \hat{\mathbf{\Psi}}\right\|$ to measure the rate of convergence. As the underlying norm of convergence for word embeddings is not known in advance, we report the quantitative results under multiple norms; ℓ_1, ℓ_2, ℓ_F and ℓ_F (Frobenius norm). Since these exponential embedding models are usually evaluated using cosine similarity metric, we also report mean cosine similarity between rows of the estimated matrix and the true matrix. The results are presented in Figure 6.7.

Observe how controlled sampling achieves faster convergence to the true $\mathbf{\Psi}$ under ℓ_1 and ℓ_F and noticeably fast under ℓ_2 norm. Since concreteness based sampling distribution draws samples relatively closer to the positive pair, we can interpret that negative updates possess more continuity in the embedding space in contrast to the random vector updates in the whole embedding space. These structured updates consequently have



(a) Convergence under ℓ_1

(b) Convergence under ℓ_2

(c) Convergence under ℓ_F

(d) Average cosine similarity

Figure 6.7: Convergence rates to the true Ψ matrix. We measure the error by a) ℓ_1 norm b) ℓ_2 norm c) ℓ_F Frobenius Norm d) average cosine similarity between rows of true and estimated $\hat{\Psi}$ matrices. CGSL instances reach gold standard embeddings faster.

6

a noticeably positive effect on the rate of convergence. Along with ℓ_p convergences, the results are further supplemented by Figure 6.7d, which is an invariant measure to the vector lengths, CGSL vectors point to the similar directions with true matrices hinting the learning effectiveness of the sampler.

6.5. CONCLUSIONS AND DISCUSSIONS

We have now investigated the limitations of standard low budget samplers, and overcame the unreliable approximations by introducing our global target variables acting as steady points during the optimization. These constraints eliminated inordinate sampling sequences for the word embedding problem. Furthermore, in order to tackle the context-free nature of the sampling, we introduced a local relevancy strategy based on word concreteness which allows us to obtain more representative sampling distributions. Along with reasonable practical performance on word similarity tasks, our novel sampling approach enjoyed variance reduced updates, with promising convergence rates.

For the future work, it is a promising direction to explore how global parameter M can be automatically adjusted during the optimization. Developing an automatic mechanism can provide further convenience for using CGSL. Besides, availability of concreteness datasets are also increasing in other languages such as German [159] and Dutch [160]. This motivates flexible extensions to the multilingual settings where word concreteness between languages can provide further information to the training. Finally, since we bounded our sampling sequences using a linear class of functions for ease, it is

also particularly interesting to explore more richer and sophisticated class of functions for characterizing the valid sequences.

7

MARKOV RANDOM SUITABILITY FIELD FOR WIND FARM PLANNING

7.1. ABSTRACT

Many countries aim to integrate a substantial amount of wind energy in the near future. This requires meticulous planning, which is challenging due to the uncertainty in wind profiles. In this cghapter, we propose a novel framework to discover and investigate those geographic areas that are suited for building wind farms. We combine the key indicators of wind farm investment using fuzzy sets, and employ multiple-criteria decision analysis to obtain a coarse wind farm suitability value. We further demonstrate how this suitability value can be refined by a Markov Random Field (MRF) that takes the dependencies between adjacent areas into account. As a proof of concept, we take wind farm planning in Turkey and demonstrate that our MRF modeling can accurately find promising areas and the suitability level of investment there.

7.2. INTRODUCTION

Over the last decades, our society has developed a more comprehensive understanding of environmentally-friendly approaches to energy generation, urging us to focus more on sustainable energy sources, such as wind energy. As a result, the integration of renewable energy goals into their long-term policies has been the priority of many countries. One example is the policy by the Ministry of Energy and Natural Resources of the Turkish Republic [161], which aims to attain a wind farm capacity of 20 GW by 2023.

A large-scale integration of wind farms will challenge the main power grid, which once was built without renewables. Thus, power grid operators must carefully analyze the expansion scenarios for wind farms and plan the necessary improvements to ensure that the electric power grid will not succumb to a large reflex to renewable energy.

In recent years, many studies have been conducted to evaluate potential geographic areas for wind farms [162], [163]. A subset of these studies considers only wind speed measurements as a basis for the assessment [163], [164], ignoring any economic or environmental restrictions for wind farms. Some studies propose including a list of environmental criteria for a more realistic integration [165], [166], [167], but the accuracy of these criteria-based methods depends directly on the input data, such as the wind power characteristics, which are hard to determine exactly. Inaccuracies in input parameters can propagate easily leading to imprecise modeling. Moreover, the assessments are usually carried out for each area independently, ignoring any neighboring relations, while the surrounding geographic factors and investments, play a role in deciding on the investment in a wind farm [168]. Motivated by such reasons, we have developed a novel spatially-aware model for the wind farm suitability of areas.

The remainder of this chapter is organized as follows. In Section 7.3, we construct a grid-based model on the Cartesian plane for representing a geographic area. Subsequently, we model the indicators of wind farms via fuzzy sets and obtain an initial suitability value using multiple-criteria decision analysis. In Section 7.4, we explain our Markov Random Field (MRF) approach for providing a refined spatially-aware suitability value for wind farms. To the best of our knowledge, we are the first to combine the fuzzy logic and multiple-criteria decision analysis with MRF to find promising areas for wind farms. Finally, a comprehensive case study is provided in Section 7.5. The results of the case study suggest that our wind farm suitability methodology provides a fine-grained information for a wind farm investment.

7.3. MODELING WIND FARM SUITABILITY

In this section, we adopt a grid-based reconstruction of geographic areas and quantify the key criteria involved in wind farm investment.

7.3.1. A GRID-BASED MODEL ON THE TWO-DIMENSIONAL CARTESIAN PLANE

We propose to use a two-dimensional grid-based model of equally-sized rectangles to represent the spherical geographic area under consideration. We assume that we are given a set $\mathcal{N}$ of points k in spherical-world coordinates, composed of latitude $\phi(k)$ and longitude $\lambda(k)$ values. Each $k \in \mathcal{N}$ of these spherical points is projected onto a two-dimensional plane using a linear mapping, where the horizontal $X(k)$ coordinate is ob-

tained using the degree of longitude $\lambda(k)$ of k and the vertical $Y(k)$ coordinate of point k can be computed based on the degree of the latitude $\phi(k)$:

$$X(k) - 1 = \frac{\lambda(k) + \beta_X}{\alpha_X} \tag{7.1}$$

$$Y(k) - 1 = \frac{\phi(k) + \beta_Y}{\alpha_Y} \tag{7.2}$$

where α_X (α_Y) is the scaling between the degree of longitude (latitude) and the horizontal (vertical) coordinate, and β_X (β_Y) defines the value of longitude (latitude) of the first coordinate on the two-dimensional plane[1].

7.3.2. QUANTIFYING THE ELEMENTARY CRITERIA FOR WIND FARMS

The decision to invest in a wind farm at a certain location depends on two main criteria: wind power potential and investment disincentives. The wind power generative potential of an area can be captured by indicators such as average wind speed, wind power density, and the capacity factor of a prospective wind turbine. On the other hand, disincentive indicators can include high values of land cost and altitude levels, and the proximity to urban areas.

Ideally, an investor should review all M indicators $\{r_1(k), \ldots, r_M(k)\}$ before investing in a wind farm at an area k. However, in practice, this review process is often not performed due to the difficulty in dealing with the uncertainty, vagueness, or the lack of information in the practical decision process. In this chapter, we model those indicators of a wind farm investment using fuzzy sets [170], which enables us to explicitly deal with uncertainty. Different than the Boolean logic, in which the truth value can only be the integer values 0 or 1, fuzzy logic can handle the concept of partial truth during a decision process.

We use increasing fuzzy function $\overline{F}\big(r_i(k)\big)$ in (7.3) and decreasing fuzzy function $\underline{F}\big(r_i(k)\big)$ in (7.4) to evaluate the satisfaction degree of each indicator $r_i(k)$ for a wind farm in area k. The increasing fuzzy function represents the incentive indicators, whereas the decreasing fuzzy function represents the disincentive indicators. The resulting fuzzy membership degrees take values between 0 and 1 corresponding to the unsatisfactory and full-satisfactory evaluations of an area k, respectively.

$$\overline{F}\big(r_i(k)\big) = \begin{cases} 0 & \text{if } r_i(k) < q_i, \\ \frac{r_i(k) - q_i}{p_i - q_i} & \text{if } q_i \le r_i(k) \le p_i, \\ 1 & \text{if } r_i(k) > p_i, \end{cases} \tag{7.3}$$

$$\underline{F}\big(r_i(k)\big) = \begin{cases} 1 & \text{if } r_i(k) < p_i, \\ \frac{r_i(k) - q_i}{p_i - q_i} & \text{if } p_i \le r_i(k) \le q_i, \\ 0 & \text{if } r_i(k) > q_i, \end{cases} \tag{7.4}$$

where for each indicator r_i, q_i and p_i correspond to the thresholds of unsatisfactory and full-satisfactory evaluations, respectively.

[1] Although the linear projection is not an accurate representation of the Earth's surface, the projection has the advantage of being geometrically simple and therefore is widely used [169].

7.3.3. MULTIPLE-CRITERIA DECISION ANALYSIS OF WIND FARMS

Since we have to deal and optimize for multiple fuzzy parameters, we focus on multiple-criteria decision analysis in this section.

The perspective of an investor is important when assessing the criteria for a wind farm. For instance, an investor could consider a worst-case scenario of the related indicators or could, as the other extreme, consider a best-case scenario. Following [165], [166], we employ fuzzy logic aggregation operators to allow for variability in perspective. We use the *and* $\wedge$ and the *or* $\vee$ aggregation operators to map two extreme cases of an investor's stance on multiple-criteria decisions: The *and* operator of the fuzzy membership degrees requires the satisfaction of all desired criteria, in other words, a conservative perspective when evaluating the satisfaction degrees of the related indicator:

$$\wedge(k) = \min_{1 \le i \le M} F\big(r_i(k)\big) \tag{7.5}$$

The *or* operator $\vee$ is appropriate to model a more optimistic or lenient perspective. The implementation of the *or* operator in (7.6) passes over the less satisfactory indicators:

$$\vee(k) = \max_{1 \le i \le M} F\big(r_i(k)\big) \tag{7.6}$$

Lastly, to model the perspective of an investor in between those two extreme cases, we can use a *weighted mean* operator μ in (7.7):

$$\mu(k) = \sum_{i=1}^{M} w_i F\big(r_i(k)\big) \tag{7.7}$$

where the ultimate decision is the convex combination of the satisfaction degrees of the decision indicators, such that $\sum_i w_i = 1$.

By applying this aggregation operator to each rectangle $k \in \mathcal{N}$, we obtain an *elementary suitability value* $\hat{z}_k \in [0,1]$ of an individual area bounded by that rectangle k. However, such elementary suitability values are not fully representative yet, due to reasons mentioned in the next section, where we describe a random field approach to model a more fine-grained *spatially-aware* suitability.

7.4. SPATIAL SUITABILITY MODELING WITH MARKOV RANDOM FIELD

The suitability values $\hat{z}_k$ calculated in the previous section provide an initial suitability estimate for a rectangle k in the grid-based model. However, we have to take the following into account: Firstly, input parameters to calculate an elementary suitability can exhibit significant measurement noise and the parameters related to wind energy potential can deviate due to inaccurate measuring instruments [171]. Secondly, the proposed elementary suitability value may not be unique, since different degrees of freedom exist in the specification of the decision making process. Lastly, the construction of a grid-based model requires the projection of a geolocation onto the Cartesian plane, introducing quantification errors that must be dealt with. Motivated by these reasons, in this

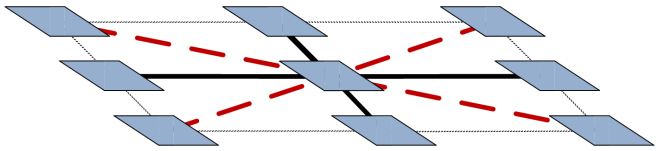


Figure 7.1: Different Markov blanket neighborhoods for the center node. Black links indicate a 4-node neighborhood, and black + red links represent an 8-node neighborhood.

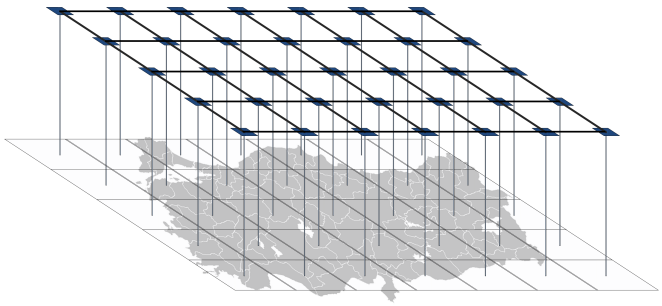


Figure 7.2: An illustration of a Markov Random Field over the grid-based model of Turkey.

section, we refine the computed elementary suitability values $\hat{z}_k$ by modeling the *true* suitability values in a MRF.

We first assume that there exist some underlying unobserved suitability values $\boldsymbol{x} = \{x_1, x_2, ..., x_k, ..., x_N\}$, but we observe a noisy version of them: $\hat{\boldsymbol{z}} = \{\hat{z}_1, \hat{z}_2, ..., \hat{z}_k, ..., \hat{z}_N\}$. Thus, the correlation among x_k and $\hat{z}_k$ is expected to be high. Let $\mathscr{B}(k)$ be the *Markov blanket* of a node k.[2] Different Markov blanket functions are shown in Figure 7.1. Then, if we assume the conditional independence of x_k, given the suitability values of the neighboring nodes $\mathscr{B}(k)$, x_k can take a value independently of the other nodes: $\mathscr{N} \setminus \mathscr{B}(k)$.

For our scenario, we adopt a 4-node neighborhood, and denote the Markov blanket of k as $\mathscr{B}_4(k)$. This defines a Markov Random Field over our grid-based model. Figure 7.2 illustrates a Markov Random Field modeling for Turkey. Horizontal links are pairwise interactions between nodes and vertical links represent the terms that force similarity between observed elementary and unknown spatially-aware suitability values.

Due to the conditional independence properties of the Markov blanket, we can write the likelihood $p(\hat{\boldsymbol{z}}|\boldsymbol{x})$ as

$$p(\hat{\boldsymbol{z}}|\boldsymbol{x}) = \prod_{k=1}^{N} \prod_{j=1}^{S} p(\hat{z}_k|x_k = j)^{\mathbb{1}_j(x_k)} \tag{7.8}$$

where $p(.)$ is the probability function, S is the number of discrete suitability states a node can take, and $\mathbb{1}(x_k)$ is the indicator vector, where all components are zero except for component x_k, which is one.

To measure the suitability state compatibility between neighboring nodes in the graph, we specify a prior, data-independent, rule. Deciding this measure of compatibility for all

[2] Rectangle k refers to an element of the grid-based model on the two-dimensional Cartesian plane, whereas node k refers to the corresponding element of the Markovian graph over our grid-based model (See Figure 7.2).

links of the graph defines the marginal probability $p(\boldsymbol{x})$. For this purpose, we use the Ising model [172], which penalizes the state incompatibility between different nodes:

$$p(\boldsymbol{x}) = \prod_{k=1}^{N} \prod_{s \in \mathscr{B}_4(x_k)} \psi(x_k, x_s) \tag{7.9}$$

where potential ψ encourages smoother solutions by forcing x_s and x_k to be in the same suitability state configuration:

$$\psi(x_k, x_s) = \frac{\gamma \exp\left(-\left|x_k - x_s\right|\right)}{G} \tag{7.10}$$

where G is the normalization term that sums over all possible state configurations of $\{k, s\}$ ensuring that $\psi(x_k, x_s)$ probabilities sum to one, and γ is the smoothness factor. The smoothness factor controls the strength of the imposed prior. For instance, $\gamma = 0$ corresponds to using no prior at all.

Adopting the pairwise interaction model along with Ising Priors, our goal is to maximize the posterior of $p(\boldsymbol{x}|\hat{\boldsymbol{z}})$, which can be computed with the help of Bayes theorem:

$$p(\boldsymbol{x}|\hat{\boldsymbol{z}}) = \frac{p(\hat{\boldsymbol{z}}|\boldsymbol{x})\, p(\boldsymbol{x})}{p(\hat{\boldsymbol{z}})} \tag{7.11}$$

Since we maximize over $\boldsymbol{x}$, we ignore the term $p(\hat{\boldsymbol{z}})$ and write the Maximum A Posteriori (MAP) estimate $\mathbf{x}_{\text{MAP}}$ of $\mathbf{x}$ as:

$$\mathbf{x}_{\text{MAP}} = \arg\max_{\boldsymbol{x}} p(\hat{\boldsymbol{z}}|\boldsymbol{x})\, p(\boldsymbol{x}) \tag{7.12}$$

Such a spatial modeling through MRF combines the local elementary suitability estimates $\hat{\boldsymbol{z}}$ to achieve globally-consistent suitability values $\boldsymbol{x}$.

To derive MAP suitability estimates, (7.12) has to be maximized: a brute-force search is out of the question even for medium-sized problems, where N is in the order of hundreds, since S discrete suitability states lead to S^N different configurations. Thus, for the solution of the multi-state case ($S > 2$), an exact MAP solution is often not applicable.[3] Then, we can resort to an Iterated Conditional Modes (ICM) algorithm for finding the MAP solution. ICM uses a greedy strategy to find the local maximum of (7.12). The idea can be stated as follows: the algorithm starts with an initial estimate of the suitability values, and then for each node $k \in \mathcal{N}$, the state configuration that gives the highest increase in the posterior probability is chosen to be the current state. This suitability-state-update procedure is continued until there are no changes in the state configuration of the nodes. This convergence is guaranteed by the ICM algorithm [175]. Even for problems with many states and nodes, the convergence of the ICM algorithm is fast, since the convergence rate is linear in S and N.

[3] Although performing maximization over the general random fields is shown to be NP-Hard [173], in the binary problem case, i.e., when $S = 2$, it was shown that maximization in (7.12) can be treated as a combinatorial maximum-flow minimum-cut problem on a graph [174].

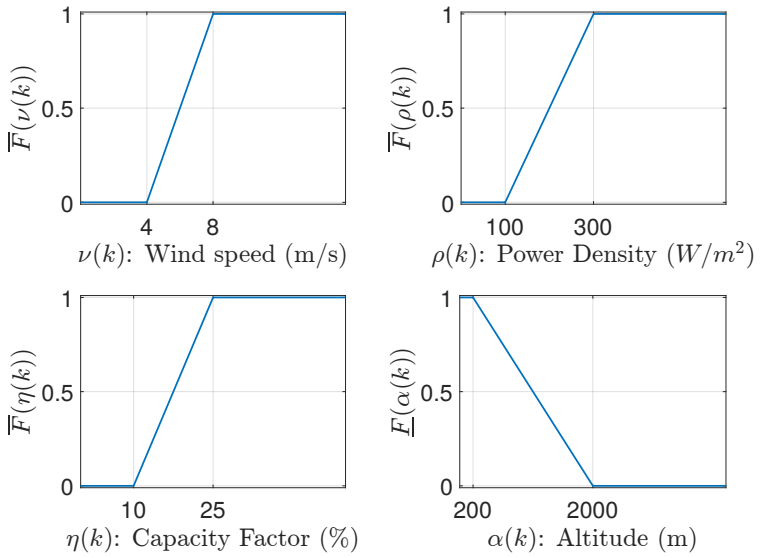


Figure 7.3: The membership functions of the selected indicators of wind farms.

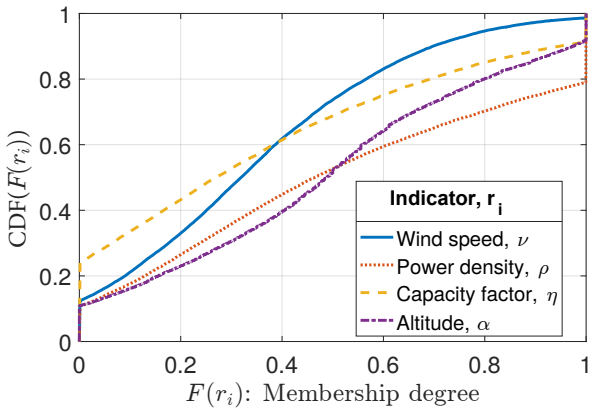


Figure 7.4: The CDFs of the membership degrees of selected indicators in Turkey.

7.5. CASE STUDY

In this section, we present a case study to demonstrate the merits of our framework. We obtained country-wide wind data of Turkey. Additionally, we collected the geographic locations in Turkey where licenses for wind farm construction are held by an investor. More details on our data collection procedure can be found in [176].

7.5.1. A GRID-BASED MODEL OF TURKEY

Based on our wind measurement data set, each rectangle in the grid-based model corresponds to a 6 km × 6 km area. As the length of a degree of latitude does not change (approximately 111.2 km); the scaling factor α_Y of vertical coordinates to a degree of latitude in (7.2) is taken as $\frac{6}{111.2} \simeq 0.053$. On the other hand, the length $l_\lambda(\phi')$ of a degree of a longitude depends on its degree of a latitude ϕ' and can be approximated as

$$l_\lambda(\phi') = \cos(\phi') \times 111.3 \text{ km}. \tag{7.13}$$

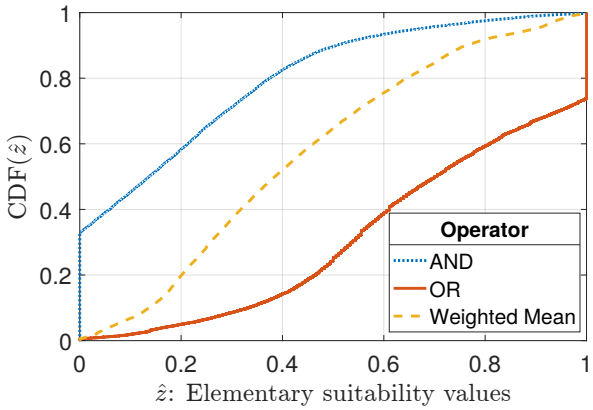


Figure 7.5: The CDFs of elementary suitability values in Turkey.

Using (7.13), around the southern points of Turkey at 36°, the length of a degree of a longitude is approximately 90 km. For ease of demonstration, in this study, we set the length of a degree of a longitude at 90 km, thus, the scaling factor α_X of horizontal coordinates to a degree of longitude in (7.1) is taken as $\frac{6}{90} \simeq 0.066$. For other purposes, it is possible to decrease the projection error by choosing variable lengths of a degree of a longitude.

The degrees of the latitude and longitude at the first coordinate on the two dimensional plane are defined according to the position of the geographic area. To fully enclose Turkey in our projection, we choose 43° latitude and 25° longitude as the first coordinate $(1,1)$. The final equations to construct the grid-based model onto a two-dimensional Cartesian plane are given in (7.14) and (7.15).

$$X(k) - 1 = \frac{\lambda(k) - 25^\circ}{0.066} \tag{7.14}$$

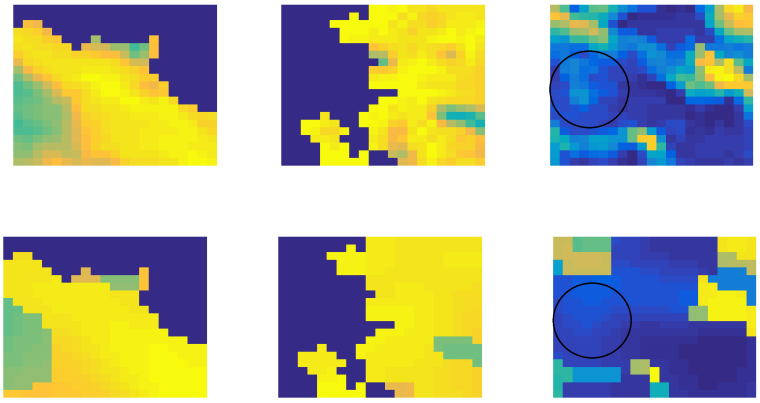
$$Y(k) - 1 = \frac{43^\circ - \phi(k)}{0.053} \tag{7.15}$$

7.5.2. QUANTIFYING THE WIND FARM POTENTIAL IN TURKEY

Key indicators [177] to capture the wind energy potential at an area k are the average wind speed $v(k)$, the wind power density $\rho(k)$, and the capacity factor $\eta(k)$ of the probable wind turbine at that area. Due to the positive correlation between the promising wind energy potential and the investment criteria for wind farms, the increasing fuzzy function in (7.3) is used to calculate corresponding satisfaction degrees of those indicators.

The landscape of Turkey contains heterogeneously distributed mountainous regions with varying altitudes. High altitude regions and high slope lands are undesirable for establishing wind farms. Thus, we use the altitude $\alpha(k)$ of a geographic area k as a disincentive indicator for wind farms. Due to the negative correlation between the altitude and the investment criteria for wind farms, the decreasing fuzzy function in (7.4) is used.

The resulting membership functions of selected indicators are shown in Figure 7.3. The full-satisfactory and unsatisfactory thresholds of indicators are determined based



(a) Tekirdağ (b) Çanakkale (c) Muş

Figure 7.6: Local patches extracted from various regions in Turkey. Patches in the first and second rows correspond to the the elementary and the spatially-aware suitability map for wind farms obtained by our MRF modeling, respectively. The dark blue coloured rectangles in (a) and (b) have zero suitability values which are outside of Turkey.

on related work [167], [176].

Next, for each area k, we calculate the membership degrees of the selected indicators for wind farms. The Cumulative Distribution Functions (CDF) corresponding to all areas in Turkey are shown in Figure 7.4. For each indicator, approximately 15% of all areas have 0 membership values, corresponding to the unsatisfactory evaluations for wind farms.

Finally, to represent the preference of an investor for the multiple-criteria decision, we use the three aggregation operators presented in Section 7.3.3. In the weighted mean operator (7.7), each of the 4 indicators is given an equal weight of 0.25. Figure 7.5 depicts the Cumulative Density Functions of the elementary suitability values with different aggregation operators. The *and* operator represents the conservative evaluation: 35% of the areas in Turkey have the minimum (0) elementary suitability value, whereas the *or* operator represents the optimistic evaluation: 25% the areas in Turkey have the maximum (1) elementary suitability value. On the other hand, the *weighted mean* operator represents a smoother evaluation: Almost none of the areas has an extreme $\{0,1\}$ elementary suitability value.

7.5.3. SPATIALLY-AWARE SUITABILITY FOR WIND FARMS IN TURKEY

We apply the Markov Random Field described in Section 7.4 to obtain the spatially-aware suitability values of wind farms in Turkey. Elementary suitability values for wind farms in Turkey are determined using the *weighted mean* operator. Our ICM algorithm visits the nodes sequentially. The number of nodes in the Markovian graph $N = 21,983$ and the number of states a node can take is set to $S = 256$.

Qualitative Results: Figure 7.6 depicts local patches extracted from distinctive regions in Turkey. The rectangles in the grid-based model are colored according to their suitability value for wind farms. Lighter colors represent higher suitability values. We observe that the suitability values for wind farms are particularly high in the Çanakkale region. This Aegean region benefits from the strong south-westerly wind, Lodos. On the

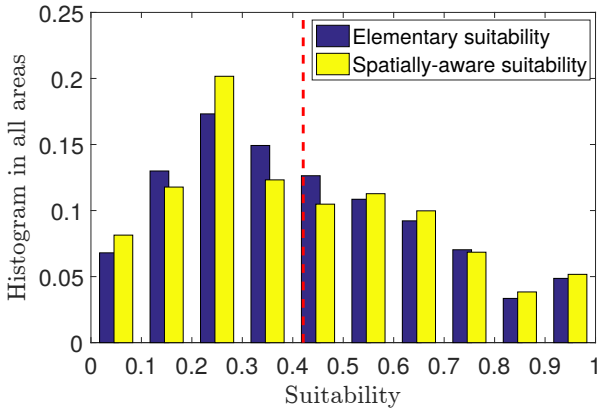


Figure 7.7: The normalized histogram of the suitability values in all geographic areas in Turkey. The red dashed line ($x = 0.42$) corresponds to the mean value of the spatially-aware suitability values of all geographic areas in Turkey.

other hand, the Muş region has lower suitability values as a consequence of its relatively high altitude and continental climate.

The spatially-aware suitability values for wind farms in the second row of Figure 7.6 include not only the individual characteristics of a specific area, but also its neighboring areas. Thus, the suitability map by MRF seems more smooth and more globally consistent. As an example, even though an area surrounded by disincentives (such as high mountains) can have higher values of elementary suitability, its spatially-aware suitability value could be lowered due to the neighboring disincentives (See Figure 7.6 (c)).

Quantitative Results: To investigate the practicability of the proposed spatially-aware suitability values for wind farms, we investigate the suitability values of licensed wind farm locations in Turkey. Figures 7.7 and 7.8 depict the histograms of the suitability values in all geographic areas and in the licensed areas for wind farms in Turkey, respectively. The spatially-aware suitability value distribution for licensed wind farm locations in Figure 7.8 follows an increasing behavior. In particular, most of the licensed wind farm locations have high suitability values. However, there are few regions where the suitability values are extraordinarily small. The calculated suitability values are insufficient for a full explanation of the license acquisition behavior in those regions. These might be overcome by augmenting more socio-economic indicators in the suitability analysis.

Next, we compare the elementary and spatially-aware suitability values of the licensed geographic locations of wind farms to analyze whether a spatially-aware suitability modeling with MRF captures additional clues about the investing behavior. Using all the regions, we first compute the expected $\mathrm{E}[\boldsymbol{x}]$ suitability of all areas. The expected $\mathrm{E}[\boldsymbol{x}]$ suitability of Turkey, the vertical dashed-red line in Figure 7.7, corresponds to the suitability for a wind farm given that an investor made a random choice for a geographic area.

Subsequently, we calculate the *tail probability* F_T in (7.16) that is the license acquisition event of an area whose suitability is smaller than the expected $\mathrm{E}[\boldsymbol{x}]$ suitability. We hypothesize that investors have access to a diverse set of common and privileged sources of information, such that their license acquisition behavior is a measure of *true* suitabil-

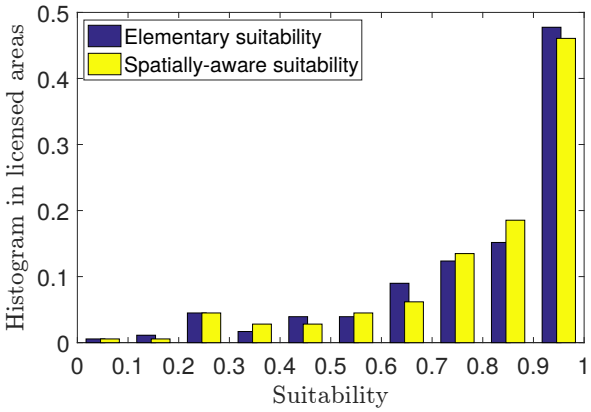


Figure 7.8: The normalized histogram of the suitability values in the licensed areas for wind farms in Turkey.

ity and they make a better decision than a random choice for an area for wind farms. Thus, we expect a lower value of tail probability F_T in the licensed geographic locations of wind farms if we can better capture the true suitability values using MRF.

$$F_T = p(x_l < \mathrm{E}[\boldsymbol{x}]) \tag{7.16}$$

where x_l is the suitability value of a licensed area.

The tail probabilities F_T in (7.16) are calculated as 0.11 and 0.08 in elementary and spatially-aware suitability models for the licensed geographic locations of wind farms. Thus, the suitability model with MRF decreases the tail probability more than 12% compared to an elementary suitability model for wind farms in Turkey. In addition to the reasons in Section 7.4, i.e., decrease in measurement noise and quantification errors, this decrease in the tail probability can be further explained by the spatial dependence of wind characteristics and warm farm investments. Investors could have a tendency to acquire a license of a region where the neighboring areas are already licensed and have a high suitability value. As a result, we can conclude that our spatially-aware modeling can help to estimate the effects of the unknown socio-economic factors on the wind farm suitability and investment decision process.

7.6. CONCLUSION

In this paper, we have proposed a framework to calculate the suitability of a geographic area for wind farms. Given the wind energy potential and the disincentive indicators of wind farms, initial suitability values for wind farms have been formed via fuzzy logic and multiple-criteria decision analysis. Subsequently, we refined those initial suitability values by a Markov Random Field model that can include the effects of the spatial relations on the wind farm suitability. Our results from a case study in Turkey show that such a spatially-aware modeling can estimate the suitability of a geographical area for wind farms accurately.

8

CONCLUSION AND DISCUSSIONS

This thesis proposes novel word embedding techniques. In this section, we summarize our contributions, discuss the potentials and limitations, and provide promising future extensions of our solutions.

Research Question 2 addressed the limited data availability issue for training word embeddings. We developed a regularized model that penalized the discrepancy between target and context embeddings. Our approach obtained a more symmetric powered decomposition than Skip Gram. Recently, [178] suggested that these context and target matrices are noisy copies of each other. This point of view gives another interpretation of our method. Our discrepancy regularization term might be interpreted as a noise reduction. This view is supported by our sensitivity analysis where we saw that when training samples are randomly dropped out, the regularized versions are less affected.

In *Research Question 1*, our hypothesis was that we can automatically distinguish different word senses from the given context. For this purpose, we derived a new embedding model with a novel mixture-based objective function to take polysemy into account and is able to distinguish different senses of words automatically. We were able to achieve better performance on word similarity datasets, such as the SCWS dataset where many query pairs consist of polysemous words. Consequently, polysemy relations can be automatically discovered while retaining the speed of a neural word embedding architecture. This preserves the scalability of our embedding model and enables future extensions to large datasets.

Fully addressing the polysemy accurately with zero supervision might, however, be problematic, certainly, when there are many polysemous words (e.g. Semitic languages). Further work in polysemy modelling should seek whether it is possible to relax the zero supervision assumption and include a tolerable amount of expert annotation. We believe that this would be the direction for a more robust and generalizable modelling for the polysemy phenomenon.

In *Research Question 3*, we investigated a methodology for fusing the lexical sources with varying structure in order to specialize embeddings for the semantics. Our method first highlighted the weakness of Distributional Hypothesis when specializing to the se-

mantic structure of the sentences and then introduced soft and hard constraints for mixing data sources of different informativeness and data availability. Finally, a refining step of bidirectional constraint flows between light and heavy sets increased the overall reliability of our light constraints, and increased the number of heavy constraints.

Our experimental results showed that semantic specialized vectors achieved much higher scores on almost all word similarity tasks. Perhaps more importantly, leveraging different informativeness of semantic sources yielded a much more stable embedding model in which word neighbours do not change arbitrarily when varying training instances. This suggests that it is difficult for traditional embedding approaches to handle the syntactic information together with semantic information in the sentences automatically. Hence, we conclude that the syntactic structure in sentences should be treated separately if we are to obtain stable word embeddings.

In our semantic learning framework, we treated all the dictionary sources equally. However, a more sophisticated embedding model can automatically optimize the contribution weights of lexical dictionaries for a semantic task at hand. Since the performance improvements we obtain with our method are quite promising, applying our semantic embedding model to other languages is a direction we suggest for further investigation, especially to the languages with richer syntactic structure.

We provided answers to *Research Question 2* and *Research Question 3* by proposing models complying with the varying amount of available training data. Although obtaining more and more data is helpful for learning of the word embeddings, additional solutions based on attention mechanism of text can supplement the performance of models including our semantics specialized embedding model.

In *Research Question 4* we do not ground on variations on the training data size, but rather question the possible improvements on the learning phase of the word embedding model. For this purpose, we sought a principled approach to efficiently optimize the negative sampling distributions, rather than following the commonly used heuristic specifications.

Our reformulation of the word embedding objective function expressed the optimization with a physical analogy, where negative sampling acts as a repulsive force. The optimization of the negative sampling distribution then boiled down to optimizing the repulsion term where we follow the the Maximum Entropy principle. Following this principle for the optimization, we made the least amount of assumptions to find the negative sampling distribution. Our surrogate quadratically constrained maximum entropy approach then posed a convex and computationally attractable solution. Since the technique has linear time complexity with respect to the vocabulary size, it permits scaling to the large-scale word embedding problems.

The performance gain was most perceivable for word similarity datasets composed of rare words which are sparsely present in the training set. We conclude that joint optimization of the model and negative distribution is indispensable if we are given new word similarity test datasets. Interestingly, the extra optimization of the negative distribution implied much faster convergence rates as shown by our synthetic experiments, which was also our intention with.

In *Research Question 5*, our purpose was to develop a negative sampler for learning embeddings much faster. We highlighted that the negative sampling approach is a tech-

nique to approximate the partition function of the exponential word embedding model. However, unlike maximum likelihood based estimators, the negative sampling approximation is designed for reducing the training time to a manageable amount. Empirical evidence of our study showed that when the sampling budget is not large, there exists an approximation gap for the negative sampler, which can be viewed as a quantity that we can correct. To imitate the intractable but true partition function, we aimed to minimize this gap online via our guided sampling mechanism. Since we eliminated unfit sampling candidates generated by the negative sampling quickly during the training, this strategy resulted in reducing the sampling variance and enabled faster convergence to gold standard embeddings. Promising future direction in this work is the investigation of richer function classes to exploit occurring sampling patterns to obtain further speed-ups.

In order to address *Research Question 6*, we proposed MRF for spatially-aware wind farm planning and obtained insights about which potential wind farms to use for Turkey. Assuming that human investors have privileged information on wind farm planning, our model had better alignment with investor decisions. This agreement shows that not only the noise in text but also measurement noise in wind energy scenarios can be efficiently dealt with random field modelling. Since the aforementioned approach is found to be so effective in this particular application, we also raise the question of whether spatial-awareness property of MRF can benefit word representations even further. For instance, word usage changes in neighboring regions can be characterized by such spatial modeling in future work.

8.1. FUTURE RESEARCH

In this section, we briefly elaborate and discuss some future directions. For many embedding models achieving state of the art performance, the number of embedding dimensions is set to very high. The intrinsic motivation for this design is that we can't accurately represent distances arising from nodes having multiple neighbors in a few dimensional spaces. Higher dimensional spaces give us the essential freedom for representing such relations. But, do we really need to train models with so many dimensions? Is it possible to adjust the capacity of the model dynamically? Providing a reasonable answer to this capacity problem would not only reduce the possible risks of overfitting but also reduce the amount of time spent on training the embeddings.

In the introduction section, we have touched upon the difficulty of a formal definition of the meaning of a word, which complicates the evaluation of word embeddings severely. Consequently, there exist no ground truth data. We advocate that the inherent subjectivity of word meaning needs to be addressed fundamentally. This is hard since even humans will disagree based on cultural differences, personal beliefs, and even that state of mind when annotating.

For one, we believe that the size of the word similarity measurement sets should be enlarged both in the dimension of the number of words and the number of experts. If we are to compromise between collecting more query word pairs and asking more annotations from experts, we believe that collecting more expert opinions is going to be relatively more helpful. But we need to take into account the background of the experts to avoid diversity biases. Hence, it is necessary to include expert profiles into the datasets, to reassure the objectivity in the end.

The embedding models in this thesis were primarily developed and tested for the English language. There are two primary reasons for this. First, there are plenty of publicly available text data on the internet in English. Second, English enjoys a clear standardisation of the language [179]. But, the native speakers of English represent only 5.52% of the world population [180]. Consequently, we do not leverage the vast information available in other languages. When we want to obtain word embeddings for other languages, it turns out that such an extension is not straightforward for many reasons. There is often much fewer data available. But also, other languages often have a much more complex grammatical and morphological systems which yield the per-word embedding representations insufficient.

For example, within the families of both Germanic and Indo-European languages, there exist some forms, such as cases, injections, and conjugations that are much more prevalent than in comparable families [181]. And, this grammatical gap increases even more radically if we look beyond Germanic or Indo-European languages.

One such example is the Malay language, in which derivational morphology is extensively used. Also, new words can be formed through axes, compounding or reduplication. We foresee that naively using these words as a basic unit will pose problems for current embedding representations. Interestingly, Bojanowski et al. [113] generalize the word level embeddings to sub-word representations which is potentially a more appropriate model for developing embeddings for morphology-rich languages. But, it is still an open question of how to design grammar and morphology aware priors for these sub-word representations for obtaining more appropriate embeddings for the language of interest.

Another future prospect is the exploration and quantification of the reliability of the training set samples. We believe that this is especially critical for multi-author content text corpora. For instance, text corpora such as Wikipedia consists of thousands of articles where virtually any user can sign up and are allowed to create or modify with little or no constraints. It is, however, impossible to assign reliability levels to this gigantic amount of text using human annotators. It would be helpful to at least know what is the minimal human labor limit on the amount of regulation such sources should exhibit.

Alternatively, we may opt for weaker reliability requirement, and associate the reliability of a text source with the bias of its author. We may utilize the author profile and page modification statistics in order to obtain a biased estimate for the articles and quantify the overall bias of the system. There might be even dependencies between authors, leading to system-wide biases. For example, one can claim that page modifications might be well-organized, and collaborative long term efforts of particular user groups [182]. Given biased sources, and author dependency estimates, bounding the overall bias changes would provide a useful answer for quantifying the reliability of text sources.

REFERENCES

[1] J. Locke, *An essay concerning human understanding book III: Of words,* (1690).

[2] L. Wittgenstein, *Blue and brown books,* (1958).

[3] M. H. Christiansen and S. Kirby, *Language evolution: the hardest problem in science?* Studies in the Evolution of Language **3**, 1 (2003).

[4] *Global internet usage,* https://en.wikipedia.org/wiki/Global_Internet_usage ().

[5] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, *Feature-rich part-of-speech tagging with a cyclic dependency network,* in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (Association for Computational Linguistics, 2003) pp. 173–180.

[6] A. Bordes, J. Weston, and N. Usunier, *Open question answering with weakly supervised embedding models,* in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, 2014) pp. 165–180.

[7] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, *Learning word vectors for sentiment analysis,* in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11 (2011) pp. 142–150.

[8] J. A. Rodriguez-Serrano, F. Perronnin, and F. Meylan, *Label embedding for text recognition,* in *Proceedings of the British Machine Vision Conference* (Citeseer, 2013).

[9] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, *Show and tell: A neural image caption generator,* CoRR **abs/1411.4555** (2014), arXiv:1411.4555 .

[10] K. Sharma, A. C. Kumar, and S. M. Bhandarkar, *Action recognition in still images using word embeddings from natural language descriptions,* in *2017 IEEE Winter Applications of Computer Vision Workshops (WACVW)* (2017) pp. 58–66.

[11] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, *Deep learning with word embeddings improves biomedical named entity recognition,* Bioinformatics **33**, i37 (2017).

[12] E. Asgari and M. R. K. Mofrad, *Continuous distributed representation of biological sequences for deep proteomics and genomics,* PLOS ONE **10**, 1 (2015).

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, *A neural probabilistic language model,* J. Mach. Learn. Res. **3** (2003).

[14] V. V. Bochkarev, A. V. Shevlyakova, and V. D. Solovyev, *Average word length dynamics as indicator of cultural changes in society,* CoRR **abs/1208.6109** (2012), arXiv:1208.6109 .

[15] J. R. Firth, *A synopsis of linguistic theory 1930-55. Studies in Linguistic Analysis (special volume of the Philological Society),* **1952-59**, 1 (1957).

[16] M. Sahlgren, *The distributional hypothesis,* Italian Journal of Linguistics **20**, 33 (2008).

[17] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, *Curriculum learning,* in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09 (ACM, New York, NY, USA, 2009) pp. 41–48.

[18] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, *Automated curriculum learning for neural networks,* CoRR **abs/1704.03003** (2017), arXiv:1704.03003 .

[19] O. Levy and Y. Goldberg, *Neural word embedding as implicit matrix factorization,* in *Advances in Neural Information Processing Systems 27* (2014) pp. 2177–2185.

[20] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, *From word embeddings to document distances,* in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15 (JMLR.org, 2015) pp. 957–966.

[21] A. Kutuzov, M. Kopotev, T. Sviridenko, and L. Ivanova, *Clustering comparable corpora of russian and ukrainian academic texts: Word embeddings and semantic fingerprints,* CoRR **abs/1604.05372** (2016), arXiv:1604.05372 .

[22] S. Arora, Y. Liang, and T. Ma, *A simple but tough-to-beat baseline for sentence embeddings, International Conference on Learning Representations,* (2017).

[23] Q. Le and T. Mikolov, *Distributed representations of sentences and documents,* in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14 (JMLR.org, 2014) pp. II–1188–II–1196.

[24] G. Ji, R. Bamler, E. B. Sudderth, and S. Mandt, *Bayesian paragraph vectors,* CoRR **abs/1711.03946** (2017), arXiv:1711.03946 .

[25] F. Hill, K. Cho, and A. Korhonen, *Learning distributed representations of sentences from unlabelled data,* CoRR **abs/1602.03483** (2016), arXiv:1602.03483 .

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent dirichlet allocation,* J. Mach. Learn. Res. **3**, 993 (2003).

[27] T. S. Jaakkola, *Tutorial on variational approximation methods,* in *IN ADVANCED MEAN FIELD METHODS: THEORY AND PRACTICE* (MIT Press, 2000) pp. 129–159.

[28] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, *Fast collapsed gibbs sampling for latent dirichlet allocation,* in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08 (ACM, New York, NY, USA, 2008) pp. 569–577.

[29] T. L. Griffiths and M. Steyvers, *Finding scientific topics,* Proceedings of the National Academy of Sciences **101**, 5228 (2004).

[30] D. M. Blei and J. D. Lafferty, *Correlated topic models,* in *In Proceedings of the 23rd International Conference on Machine Learning* (2006) pp. 113–120.

[31] B. Zhou, X. Wang, and X. Tang, *Random field topic model for semantic region analysis in crowded scenes from tracklets,* in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011).

[32] T. Hospedales, S. Gong, and T. Xiang, *A markov clustering topic model for mining behaviour in video,* in *Computer Vision, 2009 IEEE 12th International Conference on* (2009).

[33] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum, *Integrating topics and syntax,* in *In Advances in Neural Information Processing Systems 17*, Vol. 17 (2005) pp. 537–544.

[34] D. M. Blei and J. D. Lafferty, *Dynamic topic models,* in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06 (2006) pp. 113–120.

[35] R. Emonet, J. Varadarajan, and J. Odobez, *Temporal analysis of motif mixtures using dirichlet processes,* IEEE Trans. Pattern Anal. Mach. Intell. **36**, 140 (2014).

[36] M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families, and variational inference,* Found. Trends Mach. Learn. **1**, 1 (2008).

[37] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (The MIT Press, 2012).

[38] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 3rd ed. (Springer Publishing Company, Incorporated, 2009).

[39] B. Glocker, A. Sotiras, N. Komodakis, and N. Paragios, *Deformable medical image registration: Setting the state of the art with discrete methods,* Annual Review of Biomedical Engineering **13**, 219 (2011), pMID: 21568711.

[40] G. R. Cross and A. K. Jain, *Markov random field texture models,* IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-5**, 25 (1983).

[41] J. Diebel and S. Thrun, *An application of markov random fields to range sensing,* in *Advances in neural information processing systems* (2006) pp. 291–298.

[42] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, *A comparative study of energy minimization methods for markov random fields with smoothness-based priors,* IEEE transactions on pattern analysis and machine intelligence **30**, 1068 (2008).

[43] D. Chen, J.-M. Olobez, and H. Bourlard, *Text segmentation and recognition in complex background based on markov random field,* in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, Vol. 4 (IEEE, 2002) pp. 227–230.

[44] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. H. Hovy, and N. A. Smith, *Retrofitting word vectors to semantic lexicons.* CoRR (2014).

[45] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)* (Wiley-Interscience, 2006).

[46] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. (Chapman and Hall/CRC, 2004).

[47] G. Grefenstette and J. Nioche, *Estimation of english and non-english language use on the WWW,* CoRR **cs.CL/0006032** (2000).

[48] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality,* CoRR (2013).

[49] S. Clinchant and F. Perronnin, *Textual similarity with a bag-of-embedded-words model,* (2013) pp. 25:117–25:120.

[50] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning, *Bilingual word embeddings for phrase-based machine translation,* in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2013) pp. 1393–1398.

[51] D. E. Klein and G. L. Murphy, *Paper has been my ruin: conceptual relations of polysemous senses,* Journal of Memory and Language **47**, 548 (2002).

[52] D. Geeraerts, *Vagueness's puzzles, polysemy's vagaries,* Cognitive Linguistics **4**, 223 (1993).

[53] A. Tversky and J. W. Hutchinson, *Nearest Neighbor Analysis of Psychological Spaces,* Psychological Review **93**, 3 (1986).

[54] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, *A neural probabilistic language model,* Journal of Machine Learning Research **3**, 1137 (2003).

[55] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, *Natural language processing (almost) from scratch,* Journal of Machine Learning Research **12**, 2493 (2011).

[56] A. Mnih and G. Hinton, *Three new graphical models for statistical language modelling,* (ACM, 2007) pp. 641–648.

[57] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality,* CoRR **abs/1310.4546** (2013).

[58] T. Mikolov, W. tau Yih, and G. Zweig, *Linguistic regularities in continuous space word representations,* in *(NAACL-HLT-2013)* (2013).

[59] T. Kekec and D. Tax, *Robust gram embeddings,* in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2016).

[60] Q. V. Le and T. Mikolov, *Distributed representations of sentences and documents,* CoRR (2014).

[61] A. H. Michael U. Gutmann, *Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics,* Journal of Machine Learning Research **13**, 307 (2012).

[62] A. Mnih and K. Kavukcuoglu, *Learning word embeddings efficiently with noise-contrastive estimation,* in *Advances in Neural Information Processing Systems* (2013) p. 2265–2273.

[63] F. Morin and Y. Bengio, *Hierarchical probabilistic neural network language model,* in *AISTATS'05* (2005) pp. 246–252.

[64] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization,* in *NIPS* (2001).

[65] R. Lebret and R. Collobert, *Word embeddings through hellinger pca,* in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (2014).

[66] J. Li and D. Jurafsky, *Do multi-sense embeddings improve natural language understanding?* CoRR **abs/1506.01070** (2015), arXiv:1506.01070 .

[67] A. Trask, P. Michalak, and J. Liu, *sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings,* CoRR **abs/1511.06388** (2015), arXiv:1511.06388 .

[68] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, *Efficient non-parametric estimation of multiple embeddings per word in vector space,* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (2014) pp. 1059–1069.

[69] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T. Liu, *A probabilistic model for learning multi-prototype word embeddings,* in *COLING 2014, 25th International Conference on Computational Linguistics* (2014).

[70] J. Reisinger and R. J. Mooney, *Multi-prototype vector-space models of word meaning,* in *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2010)* (2010) pp. 109–117.

[71] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, *Improving word representations via global context and multiple word prototypes,* in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* ACL '12, Vol. 1 (2012).

[72] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, *Topical word embeddings,* in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15 (AAAI Press, 2015) pp. 2418–2424.

[73] X. Yang and K. Mao, *Learning multi-prototype word embedding from single-prototype word embedding with integrated knowledge,* Expert Systems with Applications **56**, 291 (2016).

[74] X. Chen, Z. Liu, and M. Sun, *A unified model for word sense representation and disambiguation,* in *Conference on Empirical Methods in Natural Language Processing* (2014).

[75] R. Navigli, *Word sense disambiguation: A survey,* ACM Comput. Surv. **41**, 10:1 (2009).

[76] S. Changpinyo, K. Liu, and F. Sha, *Similarity component analysis,* in *Advances in Neural Information Processing Systems* (2013) pp. 1511–1519.

[77] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent Dirichlet Allocation,* Journal of Machine Learning Research **3** (2003).

[78] L. van der Maaten and G. Hinton, *Visualizing non-metric similarities in multiple maps,* Machine Learning **87**, 33 (2012).

[79] J. Pennington, R. Socher, and C. Manning, *GloVe: Global vectors for word representation,* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014).

[80] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics (Springer New York Inc., New York, NY, USA, 2001).

[81] G. A. Miller, *Wordnet: A lexical database for english,* Commun. ACM **38**, 39 (1995).

[82] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, *Placing search in context: The concept revisited, ACM Trans. Inf. Syst.*, (2001).

[83] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, *Linear algebraic structure of word senses, with applications to polysemy,* CoRR **abs/1601.03764** (2016).

[84] M. Baroni, G. Dinu, and G. Kruszewski, *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors,* in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (2014) pp. 238–247.

[85] D. Ghosh, W. Guo, and S. Muresan, *Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words.* in *EMNLP* (The Association for Computational Linguistics, 2015) pp. 1003–1012.

[86] B. Lemaire and G. Denhière, *Effects of high-order co-occurrences on word semantic similarities,* CoRR **abs/0804.0143** (2008).

[87] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, *Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings,* CoRR **abs/1502.03520** (2015).

[88] P. D. Turney and P. Pantel, *From frequency to meaning: Vector space models of semantics,* J. Artif. Int. Res. **37**, 141 (2010).

[89] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, *Euclidean embedding of co-occurrence data,* J. Mach. Learn. Res. **8**, 2265 (2007).

[90] M. Fazel, H. Hindi, and S. P. Boyd, *A rank minimization heuristic with application to minimum order system approximation,* in *In Proceedings of the 2001 American Control Conference* (2001) pp. 4734–4739.

[91] N. Srebro and A. Shraibman, *Rank, trace-norm and max-norm.* in *COLT*, Lecture Notes in Computer Science, Vol. 3559 (Springer, 2005) pp. 545–560.

[92] J. Caron, *Experiments with LSA scoring: Optimal rank and basis,* in *Proc. of SIAM Computational Information Retrieval Workshop* (2000).

[93] J. Pennington, R. Socher, and C. Manning, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, 2014) pp. 1532–1543.

[94] M. U. Gutmann and A. Hyvärinen, *Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics,* J. Mach. Learn. Res. **13**, 307 (2012).

[95] A. Mnih and K. Kavukcuoglu, *Learning word embeddings efficiently with noise-contrastive estimation,* in *Advances in Neural Information Processing Systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (2013) pp. 2265–2273.

[96] T. Schnabel, I. Labutov, D. M. Mimno, and T. Joachims, *Evaluation methods for unsupervised word embeddings.* in *EMNLP* (2015) pp. 298–307.

[97] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, *A study on similarity and relatedness using distributional and wordnet-based approaches,* in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09 (2009) pp. 19–27.

[98] R. Lebret and R. Lebret, *Word emdeddings through Hellinger PCA,* CoRR **abs/1312.5542** (2013).

[99] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space,* CoRR **abs/1301.3781** (2013).

[100] Q. Luo, W. Xu, and J. Guo, *A study on the cbow model's overfitting and stability,* in *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning*, Web-KR '14 (ACM, 2014) pp. 9–12.

[101] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net,* Journal of the Royal Statistical Society, Series B **67**, 301 (2005).

[102] J. Firth, *A synopsis of linguistic theory 1930-1955,* Studies in linguistic analysis , 1 (1957).

[103] A. Tversky, *Features of similarity,* Psychological Review **84**, 327 (1977).

[104] Z. Wang, J. Zhang, J. Feng, and Z. Chen, *Knowledge graph and text jointly embedding,* in *The 2014 Conference on Empirical Methods on Natural Language Processing* (2014).

[105] M. Yu and M. Dredze, *Improving lexical embeddings with semantic knowledge.* in *ACL (2)* (2014) pp. 545–550.

[106] J. Tissier, C. Gravier, and A. Habrard, *Dict2vec : Learning word embeddings using lexical dictionaries,* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017* (2017) pp. 254–263.

[107] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, *Problems with evaluation of word embeddings using word similarity tasks,* CoRR **abs/1605.02276** (2016).

[108] A. B. Dieng, C. Wang, J. Gao, and J. W. Paisley, *Topicrnn: A recurrent neural network with long-range semantic dependency,* CoRR **abs/1611.01702** (2016), arXiv:1611.01702 .

[109] Y. Jiang, W. Bai, X. Zhang, and J. Hu, *Wikipedia-based information content and semantic similarity computation,* Inf. Process. Manage. **53**, 248 (2017).

[110] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, NY, USA, 2004).

[111] V. Nebot and R. Berlanga, *Finding association rules in semantic web data,* Know.-Based Syst. **25**, 51 (2012).

[112] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality,* CoRR **abs/1310.4546** (2013).

[113] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching word vectors with subword information,* arXiv preprint arXiv:1607.04606 (2016).

[114] M. Antoniak and D. Mimno, *Evaluating the stability of embedding-based word similarities,* in *Transactions of the Association for Computational Linguistics* (2017).

[115] L. van der Maaten and G. Hinton, *Visualizing data using t-SNE,* Journal of Machine Learning Research **9**, 2579 (2008).

[116] G. A. Miller and W. G. Charles, *Contextual correlates of semantic similarity,* Language and Cognitive Processes **6**, 1 (1991).

[117] E. Bruni, N. K. Tran, and M. Baroni, *Multimodal distributional semantics,* J. Artif. Int. Res. **49**, 1 (2014).

[118] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, *A word at a time: Computing word relatedness using temporal semantic analysis,* in *Proceedings of the 20th International World Wide Web Conference* (Hyderabad, India, 2011) pp. 337–346.

[119] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, *Large-scale learning of word relatedness with constraints,* in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12 (ACM, New York, NY, USA, 2012) pp. 1406–1414.

[120] H. Rubenstein and J. B. Goodenough, *Contextual correlates of synonymy,* Commun. ACM **8**, 627 (1965).

[121] M. thang Luong, R. Socher, and C. D. Manning, *Better word representations with recursive neural networks for morphology,* in *In Proceedings of the Thirteenth Annual Conference on Natural Language Learning. Tomas Mikolov, Wen-tau* (2013).

[122] D. Gerz, I. Vulić, F. Hill, R. Reichart, and A. Korhonen, *SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity,* in *EMNLP* (2016).

[123] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, *Placing search in context: The concept revisited,* in *Proceedings of the 10th International Conference on World Wide Web*, WWW '01 (ACM, New York, NY, USA, 2001) pp. 406–414.

[124] D. Yang and D. M. W. Powers, *Verb similarity on the taxonomy of wordnet,* in *In the 3rd International WordNet Conference (GWC-06), Jeju Island, Korea* (2006).

[125] J. R. Firth, *A synopsis of linguistic theory 1930-55. Studies in Linguistic Analysis (special volume of the Philological Society)*, **1952-59**, 1 (1957).

[126] C. D. Boom, S. V. Canneyt, T. Demeester, and B. Dhoedt, *Representation learning for very short texts using weighted word embedding aggregation,* Pattern Recognition Letters **80**, 150 (2016).

[127] X. Tang and X. Wan, *Learning bilingual embedding model for cross-language sentiment classification,* in *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 2 (2014) pp. 134–141.

[128] A. Bordes, S. Chopra, and J. Weston, *Question answering with subgraph embeddings,* CoRR **abs/1406.3676** (2014).

[129] A. Mogadala and A. Rettinger, *Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification.* in *HLT-NAACL* (2016) pp. 692–702.

[130] F. Vasile, E. Smirnova, and A. Conneau, *Meta-Prod2Vec: Product embeddings using side-information for recommendation,* in *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16 (ACM, New York, NY, USA, 2016) pp. 225–232.

[131] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, *Exploring the limits of language modeling,* CoRR **abs/1602.02410** (2016).

[132] M. J. Wainwright, M. I. Jordan, *et al.*, *Graphical models, exponential families, and variational inference,* Foundations and Trends® in Machine Learning **1**, 1 (2008).

[133] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, *Hankel matrix rank minimization with applications to system identification and realization,* SIAM Journal on Matrix Analysis and Applications **34**, 946 (2013).

[134] R. T. Rockafellar, *Convex analysis*, Princeton Mathematical Series (Princeton University Press, Princeton, N. J., 1970).

[135] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, *Conic optimization via operator splitting and homogeneous self-dual embedding,* Journal of Optimization Theory and Applications **169**, 1042 (2016).

[136] S. Boyd and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

[137] B. W. Silverman, *Density Estimation for Statistics and Data Analysis* (Chapman & Hall, London, 1986).

[138] M. U. Gutmann and A. Hyvärinen, *Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics,* J. Mach. Learn. Res. **13**, 307 (2012).

[139] R. Malouf, *A comparison of algorithms for maximum entropy parameter estimation,* in *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2002) pp. 1–7.

[140] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, *Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings,* CoRR **abs/1502.03520** (2015).

[141] T. Luong, R. Socher, and C. D. Manning, *Better word representations with recursive neural networks for morphology,* in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013* (2013) pp. 104–113.

[142] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, *Building end-to-end dialogue systems using generative hierarchical neural network models,* in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16 (2016) pp. 3776–3783.

[143] C. Wu, X. Shi, Y. Chen, J. Su, and B. Wang, *Improving implicit discourse relation recognition with discourse-specific word embeddings,* in *ACL (2)* (Association for Computational Linguistics, 2017) pp. 269–274.

[144] D. Mimno and L. Thompson, *The strange geometry of skip-gram with negative sampling,* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017) pp. 2873–2878.

[145] J. Andreas and D. Klein, *When and why are log-linear models self-normalizing?* in *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015* (2015) pp. 244–249.

[146] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Ed)* (Wiley, 2001).

[147] S. J. Reddi, A. Hefny, S. Sra, B. Pöczos, and A. Smola, *On variance reduction in stochastic gradient descent and its asynchronous variants,* in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15 (MIT Press, Cambridge, MA, USA, 2015) pp. 2647–2655.

[148] L. Chen, F. Yuan, J. M. Jose, and W. Zhang, *Improving negative sampling for word representation using self-embedded features,* CoRR **abs/1710.09805** (2017), arXiv:1710.09805 .

[149] A. Paivio, *Imagery and Verbal Processes* (Holt, Rinehart and Winston, 1971).

[150] T. Mikolov, S. W.-t. Yih, and G. Zweig, *Linguistic regularities in continuous space word representations,* in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)* (Association for Computational Linguistics, 2013).

[151] M. Brysbaert, A. Warriner, and V. Kuperman, *Concreteness ratings for 40 thousand generally known english word lemmas,* Behaviour Research Methods **46**, 904 (2014).

[152] J.-L. Durrieu, J.-P. Thiran, and F. Kelly, *Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian Mixture Models, Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, , 4833 (2012).

[153] O. Melamud, J. Goldberger, and I. Dagan, *context2vec: Learning generic context embedding with bidirectional LSTM.* in *CoNLL*, edited by Y. Goldberg and S. Riezler (ACL, 2016) pp. 51–61.

[154] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, *Learning word vectors for 157 languages,* CoRR **abs/1802.06893** (2018), arXiv:1802.06893 .

[155] C. Dyer, *Notes on noise contrastive estimation and negative sampling,* CoRR **abs/1410.8251** (2014), arXiv:1410.8251 .

[156] X. Sun, H. Wang, and W. Li, *Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection,* in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2012) pp. 253–262.

[157] H. Luo, Z. Liu, H.-B. Luan, and M. Sun, *Online learning of interpretable word embeddings,* in *EMNLP* (2015).

[158] S. Lai, K. Liu, L. Xu, and J. Zhao, *How to generate a good word embedding?* CoRR **abs/1507.05523** (2015), arXiv:1507.05523 .

[159] M. Köper and S. S. im Walde, *Automatically generated affective norms of abstractness, arousal, imageability and valence for 350000 German lemmas,* in *LREC* (2016).

[160] M. Brysbaert, M. A. Stevens, S. D. Deyne, W. Voorspoels, and G. Storms, *Norms of age of acquisition and concreteness for 30,000 Dutch words.* Acta psychologica **150**, 80 (2014).

[161] Y. A. Kaplan, *Overview of wind energy in the world and assessment of current wind energy policies in turkey,* Renewable and Sustainable Energy Reviews **43**, 562 (2015).

[162] X. Lu, M. B. McElroy, and J. Kiviluoma, *Global potential for wind-generated electricity,* Proceedings of the National Academy of Sciences **106**, 10933 (2009).

[163] A. Ucar and F. Balo, *Evaluation of wind energy potential and electricity generation at six locations in turkey,* Applied Energy **86**, 1864 (2009).

[164] A. N. Celik, *A statistical analysis of wind power density based on the Weibull and Rayleigh models at the southern region of Turkey,* Renewable energy **29**, 593 (2004).

[165] D. Latinopoulos and K. Kechagia, *A GIS-based multi-criteria evaluation for wind farm site selection. A regional scale application in Greece,* Renewable Energy **78**, 550 (2015).

[166] Y. Noorollahi, H. Yousefi, and M. Mohammadi, *Multi-criteria decision support system for wind farm site selection using GIS,* Sustainable Energy Technologies and Assessments **13**, 38 (2016).

[167] N. Y. Aydin, E. Kentel, and S. Duzgun, *GIS-based environmental assessment of wind energy systems for spatial planning: A case study from western Turkey,* Renewable and Sustainable Energy Reviews **14**, 364 (2010).

[168] K. Ek and L. Persson, *Wind farms: Where and how to place them? a choice experiment approach to measure consumer preferences for characteristics of wind farm establishments in Sweden,* Ecological economics **105**, 193 (2014).

[169] L. M. Bugayevskiy and J. Snyder, *Map projections: A reference manual* (CRC Press, 1995).

[170] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*, Vol. 4 (Prentice hall New Jersey, 1995).

[171] K. Z. Østergaard, P. Brath, and J. Stoustrup, *Estimation of effective wind speed,* in *Journal of Physics: Conference Series*, Vol. 75 (IOP Publishing, 2007) p. 012082.

[172] S. Geman and D. Geman, *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,* IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721 (1984).

[173] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts,* IEEE Trans. Pattern Anal. Mach. Intell. **23**, 1222 (2001).

[174] V. Kolmogorov and R. Zabin, *What energy functions can be minimized via graph cuts?* IEEE transactions on pattern analysis and machine intelligence **26**, 147 (2004).

[175] J. Besag, *On the statistical analysis of dirty pictures,* Jounal of the royal statistical society B **48**, 48 (1986).

[176] H. Cetinay, F. A. Kuipers, and A. N. Guven, *Optimal siting and sizing of wind farms,* Renewable Energy **101**, 51 (2017).

[177] T. Ackermann, *Wind power in power systems* (John Wiley & Sons, 2005).

[178] D. Mimno and L. Thompson, *The strange geometry of skip-gram with negative sampling,* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017) pp. 2873–2878.

[179] D. Denison and R. Hogg, *A history of the English language* (Cambridge University Press, United Kingdom, 2006).

[180] *Nationalencyklopedin,* (1980).

[181] M. Haspelmath, M. S. Dryer, D. Gil, and B. Comrie, *The world atlas of language structures online,* (2013).

[182] *US congress edits to wikipedia,* https://www.theguardian.com/technology/2014/jul/25/us-congress-banned-editing-wikipedia-trolling ().

[183] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning* (The MIT Press, 2012).

A

APPENDIX A

A.1. VARIATIONAL BAYES FOR LDA

Full posterior of LDA is $p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta})$. In order to obtain model evidence (also called log marginal likelihood), we first integrate over model parameters $\boldsymbol{\theta}$ and sum over hidden variables $\boldsymbol{z}$ [1]

$$\log p(\boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta}) = \log \int \sum_{z} p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta}) d\boldsymbol{\theta} \tag{A.1}$$

We will use a variational distribution $q(\boldsymbol{\theta}, \boldsymbol{z}|\phi, \psi)$ to match this marginal likelihood. By introducing this function on nominator and denominator, we obtain:

$$\log p(\boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta}) = \log \int \sum_{z} \frac{q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega)\, p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta})}{q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega)} d\boldsymbol{\theta} \tag{A.2}$$

Since all individual factors in the LDA posterior is in exponential family distributions yielding LDA posterior to be in exponential family also. Also we make the assumption that any chosen q distribution will be also in the exponential family. The resulting $\log p(\boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta})$ will be a convex function and we can apply Jensen's inequality:

$$\begin{aligned} \log p(\boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta}) &\geq \int \sum_{z} q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega) \log \frac{p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta})}{q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega)} d\boldsymbol{\theta} \\ &\geq \int \sum_{z} q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega) \log p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta}) d\boldsymbol{\theta} - \int \sum_{z} q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega) \log q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \Omega) d\boldsymbol{\theta} \end{aligned} \tag{A.3}$$

We now see that RHS is the lower bound over $\log p(\boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta})$. The first term of RHS is recognized as the KL Divergence between the q and p distributions. The second term is recognized as the entropy functional of q function. In the best case, this bound can be equal to $\log p(\boldsymbol{w}|\boldsymbol{a}, \boldsymbol{\beta})$ which indicates a perfect variational fit. For simplification, lets

[1] Since the posterior over documents factorizes and latent variables are document-specific, we carry the derivation for only one document.

A

alias right hand side of the equation as the Lower Bound $L(\psi, \phi | \boldsymbol{a}, \boldsymbol{\beta})$. If we expand this lower bound, we will obtain a set factors each having an expectation under q:

$$L(\gamma, \Omega | \boldsymbol{a}, \boldsymbol{\beta}) = E_q[\log p(\boldsymbol{\theta}|\boldsymbol{a})] + E_q[\log p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta})] + E_q[\log p(\boldsymbol{z}|\boldsymbol{\theta})] + H(q) \tag{A.4}$$

Now, after learning variational parameters, we can query the probability of a given data point which we could not do in the original posterior. Finding best variational parameters boils down to the following optimization problem:

$$(\gamma^*, \Omega^*) = -\arg\min_{\gamma, \Omega} L(\gamma, \Omega | \boldsymbol{a}, \boldsymbol{\beta}) \tag{A.5}$$

A.2. LOWER BOUND

The lower bound for the posterior under Variational Bayes can be given as follows:

$$L(\gamma, \Omega | \boldsymbol{a}, \boldsymbol{\beta}) = \underbrace{\mathbb{E}\left[\log p(\theta|\alpha)\right]_q}_{\text{T1}} + \underbrace{\mathbb{E}\left[\log p(\mathbf{z}|\theta)\right]_q}_{\text{T2}} + \underbrace{\mathbb{E}\left[\log p(\mathbf{w}|\mathbf{z}, \beta)\right]_q}_{\text{T3}} + \underbrace{\mathbb{H}[Q]}_{\text{T4}} \tag{A.6}$$

where T1 is

$$\sum_{k=1}^{K} (a_k - 1)\, \mathbb{E}[\log \theta_k]_Q + \log\left(\Gamma(\sum a_k)\right) - \sum \log\left(\Gamma(a_k)\right) \tag{A.7}$$

since we assumed $Q(\theta, z|\gamma, \Omega) = Q(\theta|\gamma) \prod_{n=1}^{N} Q(z_n|\Omega_n)$ (Mean Field assumption).

For $Q(\theta)$, Dirichlet distribution, we know the log expectation is DiGamma function. T1 becomes:

$$\text{T1} = \sum_{k=1}^{K} (a_k - 1) \left[F(\gamma_k) - F(\sum_{k'} \gamma_{k'}) \right] + \log\left(\Gamma(\sum a_k)\right) - \sum \log\left(\Gamma(a_k)\right) + cons \tag{A.8}$$

For the second term,

$$\mathbb{E}\left[\log p(\mathbf{z}|\theta)\right]_q = \mathbb{E}\left[\log \prod_{n}^{N} \prod_{k}^{K} \theta_k^{z_{k,n}}\right] = \mathbb{E}\left[\sum_n \sum_k z_{k,n} \log \theta_k\right]_Q \tag{A.9}$$

we see that first factor requires expectation on $Q(z|\phi)$ and second $Q(\theta|\gamma)$:

$$= \sum_n \sum_k \mathbb{E}[z_{k,n}]_{Q(z_n|\phi_n)}\, \mathbb{E}[\log \theta_k]_{Q(\theta|\gamma)} \tag{A.10}$$

Since $Q(z|\phi)$ factorizes into words (exchangability), we only have to evaluate w.r.t $Q(z_n|\phi_n)$.

$$\text{T2} = \sum_n \sum_k \phi_{n,k} \left[F(\gamma_k) - F(\sum_{k'} \gamma_{k'}) \right] \tag{A.11}$$

For the third term we have

$$\text{T3} = \mathbb{E}[\log p(\mathbf{w}|\mathbf{z},\beta)]_Q = \sum_n \sum_k \mathbb{E}[z_{k,n} \log \beta_{k,w_n}]_Q = \sum_n \sum_k \phi_{n,k} \log(\beta_{k,w_n}) \tag{A.12}$$

For the last term of entropy:

$$\mathbb{H}[\mathbb{Q}] = -\int \sum_z q(\theta, z|\gamma,\phi) \log q(\theta, z|\gamma,\phi)\, d\theta \tag{A.13}$$

The good thing of mean field variational inference here is the factorization of the entropy distribution:

$$\begin{aligned}\mathbb{H}[\mathbb{Q}] &= -\int q(\theta|\gamma)\log q(\theta|\gamma) - \sum_z q(z|\phi)\log q(z|\phi) \\ &= -\mathbb{E}\left[\log q(\theta|\gamma)\right]_Q - \mathbb{E}\left[\log q(z|\phi)\right]_Q\end{aligned}$$

$$\text{T4} = -\sum_{k=1}^{K}(\gamma_k - 1)\left[\psi(\gamma_k) - \psi(\sum_{k'}\gamma_{k'})\right] - \log\left(\Gamma(\sum_k \gamma_k)\right) + \sum_k \log\left(\Gamma(\gamma_k)\right) - \sum_n \sum_k \phi_{n,k}\log\phi_{k,n}$$

A.2.1. γ VARIATIONAL UPDATE

Variational problem is:

$$\gamma^* = \arg\min -L(\theta, z|\gamma,\phi) \tag{A.14}$$

We have contributions from T1, T2, T4:

$$\frac{\partial L}{\partial \gamma_k} = \frac{\partial\left(\sum_{k=1}^{K}(a_k-1)\left[\psi(\gamma_k) - \psi(\sum_{k'}\gamma_{k'})\right]\right)}{\partial\gamma_k} + \frac{\partial\left(\sum_n\sum_k \phi_{n,k}\left[\psi(\gamma_k) - \psi(\sum_{k'}\gamma_{k'})\right]\right)}{\partial\gamma_k} + \frac{\partial\left(-\sum_{k=1}^{K}(\gamma_k-1)\left[\psi(\gamma_k) - \psi(\sum_{k'}\gamma_{k'})\right] - \log\left(\Gamma(\sum_k\gamma_k)\right) + \sum_k\log\left(\Gamma(\gamma_k)\right)\right)}{\partial\gamma_k}$$

rewrite into:

$$\frac{\partial L}{\partial \gamma_k} = \sum_k \left[(a_k - 1) + \sum_n \phi_{k,n} - (\gamma_k - 1)\right]\left[\psi(\gamma_k) - \psi(\sum_{k'}\gamma_{k'})\right] \tag{A.15}$$

$\frac{\partial L}{\partial \gamma_k}$ should be 0 for γ_k term giving closed form θ_k update

$$\gamma_k = a_k + \sum_n \phi_{n,k} \tag{A.16}$$

A.2.2. ϕ VARIATIONAL UPDATE

Variational problem is:

$$\phi^* = \arg\min -L(\theta, z|\gamma,\phi)\ s.t. \sum_k \phi_{n,k} = 1 \tag{A.17}$$

A

We have contributions from T2, T3, T4. We write the lower bound first:

$$L_\phi = \sum_n \sum_k \left[\psi(\gamma_k) - \psi(\sum_{k'} \gamma_{k'}) \right] + \sum_n \sum_k \phi_{k,n} \log(\beta_{k,w_n}) - \sum_n \sum_k \phi_{k,n} \log \phi_{k,n} + \lambda \left(\sum_k \phi_{k,n} - 1 \right)$$

by denoting subtraction of ψ functions as $S(\psi)$ rewrite into:

$$L_\phi = \sum_n \sum_k \phi_{k,n} \left[S(\psi) + \log(\beta_{k,w_n}) - \log \phi_{k,n} \right] + \lambda_n \left(\sum_k \phi_{k,n} - 1 \right) \tag{A.18}$$

Although I omit in the notation, each L_{ϕ_n} contributes $|w_n|$ times to the loss. Now, the gradient of this bound becomes::

$$\frac{\partial L}{\partial \phi_{k,n}} = \left[S(\psi) + \log(\beta_{k,w_n}) - \log \phi_{k,n} - 1 \right] + \lambda_n$$

Since L_ϕ bound is convex, this equation has closed form solution, we can set LHS to 0 and obtain:

$$\log \phi_{k,n} = S(\psi) + \log(\beta_{k,w_n}) - 1 + \lambda_n$$

we exponentiate this and absorb the constants into normalization factor, giving us

$$\phi_{k,n} \sim \exp\left(\log \beta_{k,w_n} + S(\psi) \right)$$

Running this update and then normalizing the $\phi_{.,n}$ will complete the variational update.

B

APPENDIX B

B.1. NEGATIVE SAMPLING OBJECTIVE

The negative sampling objective is given as follows:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{p_d}\left[\ln \sigma(\boldsymbol{x};\boldsymbol{\theta})\right] + \mathbb{E}_{p_n^0}\left[\ln(1-\sigma(\boldsymbol{y};\boldsymbol{\theta}))\right] \tag{B.1}$$

Here, σ is the sigmoid function:

$$\sigma(\boldsymbol{u};\boldsymbol{\theta}) = \frac{1}{1+\exp\left[-G(\boldsymbol{u};\boldsymbol{\theta})\right]}$$

where G is the difference between the log likelihood of the sample under the model and the negative sampling distribution:

$$G(\boldsymbol{u};\boldsymbol{\theta}) = \ln p_m^{\boldsymbol{\theta}}(\boldsymbol{u}) - \ln p_n(\boldsymbol{u})$$

Substitution of σ and G functions gives us the following:

$$J_T(\boldsymbol{\theta}) = \mathbb{E}_{p_d}\left[\ln \frac{p_m^{\boldsymbol{\theta}}(\boldsymbol{x})}{p_m^{\boldsymbol{\theta}}(\boldsymbol{x}) + p_n(\boldsymbol{x})}\right] + \mathbb{E}_{p_n^0}\left[\ln \frac{p_n(\boldsymbol{y})}{p_m^{\boldsymbol{\theta}}(\boldsymbol{y}) + p_n(\boldsymbol{y})}\right]$$

Using logarithmic properties and expectation additivity, we decompose this objective into:

$$\begin{aligned} J(\boldsymbol{\theta}, p_n) = \mathbb{E}_{p_d}[\ln p_m^{\boldsymbol{\theta}}(\boldsymbol{x})] - \mathbb{E}_{p_d}[\ln(p_m^{\boldsymbol{\theta}}(\boldsymbol{x}) + p_n(\boldsymbol{x}))] \\ -\mathbb{E}_{p_n^0(y)}[\ln(p_m^{\boldsymbol{\theta}}(\boldsymbol{y})) + p_n(\boldsymbol{y})) + \mathbb{E}_{p_n^0(\boldsymbol{y})}[\ln p_n(\boldsymbol{y})] \end{aligned}$$

where fourth term is constant in $\boldsymbol{\theta}$. □

B.2. SMOOTHING THE DISTRIBUTION

Assume we have a probability mass function, with *ordered* entries:

$$p_1 \geq p_2 \geq ... \geq p_n > 0 \tag{B.2}$$

with $\sum_{i=1}^{n} p_i = 1$. We smooth PMF p slightly, by modifying two neighbouring probabilities with a small probability Δ_i. This defines a new PMF $\tilde{p}$, with $\tilde{p}_i = p_i - \Delta_i$, $\tilde{p}_{i+1} = p_{i+1} + \Delta_i$, and all other probabilities remain the same. The entropy change: $H(\tilde{p}) - H(p)$ can be stated as:

$$\begin{aligned}
&= -(p_i - \Delta_i)\log(p_i - \Delta_i) - (p_{i+1} + \Delta_i)\log(p_{i+1} + \Delta_i) \\
&\quad + p_i \log p_i + p_{i+1} \log p_{i+1} \\
&= -p_i(\log(p_i - \Delta_i) - \log p_i) - p_{i+1}(\log(p_{i+1} + \Delta_i) - \log p_{i+1}) \\
&\quad + \Delta_1 \log(p_i - \Delta_i) - \Delta_i \log(p_{i+1} + \Delta_i) \\
&= -p_i(\log(1 - \frac{\Delta_i}{p_i})) - p_{i+1}(\log(1 + \frac{\Delta_i}{p_{i+1}})) \\
&\quad + \Delta_i \log(p_i(1 - \frac{\Delta_i}{p_i})) - \Delta_i \log(p_{i+1}(1 + \frac{\Delta_i}{p_{i+1}}))
\end{aligned}$$

The logarithms are of the form $\log(1+x)$ for which the Taylor expansion around $x = 0$ can be used:

$$\log(1 + x) = 0 + x + O(x^2) \tag{B.3}$$

Therefore, the substitution gives:

$$\begin{aligned}
H(\tilde{p}) - H(p) &= p_i \frac{\Delta_i}{p_i} - p_{i+1} \frac{\Delta_i}{p_{i+1}} \\
&\quad + \Delta_i \log p_i - \Delta_i \frac{\Delta_i}{p_i} - \Delta_i \log p_{i+1} - \Delta_i \frac{\Delta_i}{p_{i+1}} + O(\Delta_i^2) \\
&= + \Delta_i \log p_i - \Delta_i \log p_{i+1} + O(\Delta_i^2) \\
&= \Delta_i \log \frac{p_i}{p_{i+1}} + O(\Delta_i^2) > 0
\end{aligned} \tag{B.4}$$

The first two terms cancel, the forth and the sixth are of order $O(\Delta_i^2)$, and only the third and fifth term remain. Because $p_i > p_{i+1}$, this difference between the entropies $H(\tilde{p}) - H(p)$ is larger than 0. □

B.3. POWERING THE DISTRIBUTION

Assuming we have a probability mass function, as defined in Equation (B.2). We define a power λ, $0 < \lambda < 1$, and rescale the PMF:

$$\tilde{p}_i = \frac{p_i^\lambda}{\sum_j p_j^\lambda} \tag{B.5}$$

This new distribution is more smooth when

$$\hat{\Delta}_i \leq \Delta_i \tag{B.6}$$

where $\Delta_i = p_i - p_{i+1}$ That would mean:

$$\begin{aligned} \frac{p_i^\lambda - p_{i+1}^\lambda}{\sum_j p_j^\lambda} &\leq p_i - p_{i+1} \\ p_i^\lambda - p_{i+1}^\lambda &\leq (\sum_j p_j^\lambda)(p_i - p_{i+1}) \quad \text{(B.7)} \\ p_i^\lambda - p_{i+1}^\lambda &\leq C(p_i - p_{i+1}) \quad \text{(B.8)} \end{aligned}$$

This is actually the definition of Lipschitz continuity [183]. Unfortunately, for $f(x) = x^\lambda$ where $x \in [0,1]$ and $0 < \lambda < 1$ function f is *not* Lipschitz continuous, because for very small values of x the derivative goes to infinity.

If we now assume that $\gamma < p_i < 1$, our purpose is to derive a lower bound γ for p_i such that (B.7) actually holds. First, we define the function f:

$$\begin{aligned} f(x) &= = x^\lambda \quad x \in (0,1),\ \gamma < \lambda < 1 \\ f'(x) &= \lambda x^{\lambda-1} \quad \text{is always positive} \\ f''(x) &= \lambda(\lambda-1)x^{\lambda-2} \quad \text{is always negative} \end{aligned}$$

in other words: the derivative is always positive, but each derivative becomes smaller and smaller. Because we have that $x > \gamma$, and for $h > 0$:

$$f'(\gamma) > f'(x) = \lim_{h\downarrow} \frac{f(x+h) - f(x)}{(x+h) - x} > \frac{f(x+h) - f(x)}{(x+h) - x} \quad \text{(B.9)}$$

Using $f'(x) = \lambda x^{\lambda-1}$, and rewriting gives:

$$f(x+h) - f(x) < \lambda\gamma^{\lambda-1}((x+h) - x) \quad \text{(B.10)}$$

Substitution of $x + h = p_i$ and $x = p_{i+1}$ and solving γ reads:

$$p_i^\lambda - p_{i+1}^\lambda < \lambda\gamma^{\lambda-1}(p_i - p_{i+1}) \quad \text{(B.11)}$$

Now we can identify γ using Equation (B.7):

$$\begin{aligned} \lambda\gamma^{\lambda-1} &= \sum_j p_j^\lambda \quad \text{(B.12)} \\ \gamma &= \left(\frac{1}{\lambda}\sum_j p_j^\lambda\right)^{1/(\lambda-1)} \quad \text{(B.13)} \end{aligned}$$

Now, γ gamma is lower bounded as such, powering the distribution acts as a smoother. □

SUMMARY

The digital era floods us with an excessive amount of text data. To make sense of such data automatically, there is an increasing demand for accurate numerical word representations. The complexity of natural languages motivates to represent words with high dimensional vectors. However, learning in a high dimensional space is challenging when the amount of training data is noisy and scarce. Additionally, lingual dependencies complicate learning, mostly because computational resources are limited and typically insufficient to account for all possible dependencies. This thesis addresses both challenges by following a probabilistic machine learning approach to find vectors, word embeddings, performing well under aforementioned limitations.

An important finding of this thesis is that by bounding the length of the vector that represents a word as well as penalizing the discrepancy between vectors representing different words make a word embedding robust, which is especially beneficial when noisy and little training data is available. This finding is important because it shows how current word embedding methods are sensitive to small variations in the training data. Although, one might conclude from this finding that more data is not necessary anymore, this thesis does show that training on multiple sources, such as dictionaries and thesaurus, improves performance. But, each data source should be treated carefully, and it is important to weigh informative parts of each data source appropriately.

To deal with lingual dependencies, this thesis introduces statistical negative sampling with which the learning objective of a word embedding can be approximated. There are many degrees of freedom in the approximated learning objective, and this thesis argues that current embedding strategies are based on weak heuristics to constrain these freedoms. Novel and more theoretical founded constraints are being proposed to constrain the approximations that are based on global statistics and maximum entropy.

Finally, many words in a natural language have multiple meanings, and current word embeddings do not address this because they are built on a common assumption that one vector per word representation can capture all word meanings. This thesis shows that a representation based on multiple vectors per word easily overcomes this limitation by having different vectors representing the different meanings of a word.

Taken together, this thesis proposes new insights and a more theoretical foundation for word embeddings which are important to create more powerful models able to deal with the complexity of natural languages.

SAMENVATTING

Het digitale tijdperk overlaadt ons met een overmatige hoeveelheid tekstgegevens. Om dergelijke gegevens automatisch te begrijpen is er een toenemende vraag naar nauwkeurige numerieke woordrepresentaties van documenten. Hierbij is het natuurlijk om een hoog-dimensionale representatie van woorden te kiezen vanwege de complexiteit van natuurlijke talen. Echter, het leren in een hoog-dimensionale ruimte is een uitdaging wanneer de leerset onnauwkeurig en schaars is. Bovendien maken linguïstische afhankelijkheden het leren heel moeilijk. Vooral omdat bronnen beperkt zijn en er meestal onvoldoende rekening gehouden wordt met alle mogelijke afhankelijkheden. Dit proefschrift behandelt beide uitdagingen. Het stelt voor om een probabilistische leermethode te volgen voor het vinden van de juiste vectoren om woorden te representeren, de woordinbedding, die goed presteert onder de bovengenoemde beperkingen.

Een belangrijke conclusie van dit proefschrift is dat door zowel de lengte van de woord vectoren, als het verschil tussen woord vectorenen van vergeljkbare woorden te begrenzen, woorden robuust ingebed kunnen worden. Dit is vooral voordelig wanneer de leerset onnauwkeurig en schaars is. Deze bevinding is belangrijk omdat het laat zien hoe de huidige inbeddingsmethoden gevoelig zijn voor kleine variaties in de leerset. Hoewel je uit deze bevinding zou kunnen concluderen dat meer gegevens niet meer nodig zijn, laat dit proefschrift zien dat leren aan de hand van meerdere bronnen, zoals woordenboeken en thesauri, de prestaties verbetert. Maar elke gegevensbron moet zorgvuldig worden behandeld en het is belangrijk om de informatieve delen van elke gegevensbron op de juiste manier te wegen.

Om met linguïstische afhankelijkheden om te gaan, introduceert dit proefschrift negatieve samplingstrategieën. Daarnaast worden nieuwe en meer theoretisch gefundeerde beperkingen voor de vele vrijheidsgraden in de leermethoden geïntroduceerd, die gebaseerd zijn op globale statistieken en maximale entropie. Ten slotte hebben veel woorden meerdere betekenissen. De huidige woordinbeddingsmethoden modeleren dit niet omdat ze zijn gebaseerd op de algemene aanname dat één vector per woord alle betekenissen kan bevatten. Dit proefschrift laat echter zien dat een representatie gebaseerd op meerdere vectoren per woord beter om kan gaan met de verschillende betekenissen van een woord.

Kortom, dit proefschrift biedt nieuwe inzichten en meer theoretische basis voor woordinbeddingen, die belangrijk zijn om krachtigere modellen te maken die kunnen omgaan met de complexiteit van natuurlijke talen.

ACKNOWLEDGEMENTS

I would like to acknowledge the generous support of people, helping me to develop my skills and abilities and gradually contributing to my PhD thesis.

First of all, I would like to thank to my thesis promotor Marcel Reinders for giving me the unique opportunity to perform my doctorate study in this distinguished and top research group where I got exposed to many different research topics and state of the art knowledge. I also would like to express my gratitude to Laurens van der Maaten for nominating me in his project, and providing introductory guidance during the early period of my PhD. My supervisor David M. J. Tax has spent lots of time on my conceptual and technical development. I must admit that, sometimes during the research meetings, I conceptualized him as a city watch in medieval times. Similarly to the city watch guarding the city from the incoming threats, he spent considerable effort for protecting my research from the possible threats and most likely research pitfalls. He also kindly spent time on me to show what are the necessary and optional qualifications for a modern scientist. During my course of study, I had the wonderful privilege to meet, discuss, share ideas, and collaborate with many scientists such as Robert P.W. Duin, Marco Loog, Hayley Hung, Jan Van Gemert. I was also lucky to work with David Mimno of Cornell University. It is a striking fact for me that when a researcher intends to extract information and details of a learning scenario, infinite amount of things to learn immediately pops up. The Pattern Recognition and Bioinformatics laboratory had been a learning environment for me these four years with many colleagues around.

I would like to thank; Wouter Kouw for his lust of knowledge that is difficult to go unnoticed, Jesse H. Krijthe for demonstrating his scientific attitudes, Wenjie Li for letting me be his paranymph, Görkem Saygılı for his modesty, Tom Viering for showing personal confidence. I think his voice is an mark of high personal confidence rather than a bare loud voice. Alexander Mey for demonstrating how to think like a mathematician and how to keep fit. Veronika Cheplygina for her hospitable and welcoming behaviours, Yazhou Yang for his not only silent but also benevolent nature with qualities of being a great office mate, Yancong Lin for his very unexpected and entertaining goals, Silvia Pintea for her punctuality. Abolfazl Nadi for sharing beautiful Persian poems, Tamim Abdelaal for helping my book collection, Stavros Makrodimitris for saving me from being the worst goalkeeper of Delft, Ramin Shirali for reviving my Warcraft 3 memories. Mahdi Karami for his sedate speechs. Ekin Gedik and Laura Cabrera Quiros for depicting how an ideal brother and sister should be. I can not even disentangle their names here. Yeshwanth Napolean, Ziqi Wang, Jose Vargas, Stephanie Tan, Chirag Raman for showing how to move as a team. Oğuzhan Ersoy for his hospitality, Osman Semih Kayhan for showing how one should bear strength and faith in his moments of difficulty. I must also acknowledge Osman's patience, where he kindly tolerated my desires of introducing rare Turkish words to our conversations. Hamdi Dibeklioglu for his fast-paced and energetic nature, I also thank Saskia Peters and Günay Aslan for their helps with administrative affairs.

During the period in Delft, I am going to remember the conversations I had with my friends, and I also want to mention their names here and thank them. Kasım Sinan Yıldırım for his playful and humorist off-work conversations, Hale İyicil for teaching the management in low budget scenarios, providing support during the hard moments of research and finally demonstrating her skills in efficient methods of collaboration. Mert İyicil for his considerate and insightful nature, Reyhan Aydoğan for teaching me and others how to be a master in a board game, Duygu Güroğlu for showing how adventurous experiences is an essential part of life. I presume it is quite difficult to forget the discussions we had with Francisco Jose Márquez Bonilla and Maria Jesus. I also thank Marcus Ekström for his great and strategic team-plays.

I would like to mention some of my long-lasting friends who supported me in times of difficulty;

Esra Nur Varlı for showing her networking skills, Fatih Doğan for his insights and long-lasting devotion, Hilal Kazan for calling me with a creative nickname, Salih Yıldırım for unimaginable work perseverance, Sümeyra Balcı for demonstrating rational attitudes even at times of conflicts, and Merve Gençer for showing fidelity and friendship. Mustafa Şentürk for fruitful discussions, Cengiz Özdemir for his attentive listening. Semih Yener for his dedication and willpower, Emre Karaalioğlu for being an easygoing travel mate. Ömer Kemal Adak for his enthusiastic storytelling.

The knowledge of the humankind develops over long period of times. Similarly, this study is also a result of the accumulated knowledge acquired from many teachers, implicitly contributing to the thesis. I would like to acknowledge the endless support of my former supervisor, Mustafa Ünel. Some of the highly valued advises he gave took years of personal internalization, which were indispensable for a young scientist. He introduced me the research discipline, continuous contemplation and the indispensable persistence as key quantities a researcher should possess. Elif Karslıgil for recommending me to follow a PhD program several years ago. I also thank my elementary physics teacher Gülten Öz who trusted me in my moments of failure and desperation, and gradually taught me how hard study and dedication can give birth to the success.

I would like to thank Elif Akça, for being with me during my study period. I am grateful for your care and support. I also finally would like to thank my mother Bilgen Kekeç and my father. I consider myself privileged and proud to be the son of such a hardworking and compassionate woman. Her multi-faceted efforts spanning tens of years for my growth is difficult to describe with common words. To be honest, after recognizing all the commitments she did for me, I feel seriously in debt. This thesis would not exist without her unconditional, faithful support. The following poem of Bahtiyar Vahapzade in an alternative summary for this thesis:

“Yoh men heçem
Men yalanam
Kitap kitap sözlerimin
Müellifi menim anam.”

Taygun Kekeç
April 2019, Delft, The Netherlands

CURRICULUM VITÆ

Taygun KEKEÇ

24-07-1989	Born in Istanbul, Turkey.

EDUCATION

2002–2006	High School Burak Bora Anatolian High School, Istanbul
2006–2010	BSc in Computer Engineering & Science Yildiz Technical University, Istanbul
2011–2013	MSc in Mechatronics Engineering Sabanci University, Istanbul
Thesis:	Developing object detection, tracking and image mosaicing algorithms for visual surveillance
Promotor:	Prof. dr. M. Ünel

AWARDS

2006	Ranked 1% at Turkish University Examination over 1.5M students.
2009	Ranked 3rd at Microsoft Imagine Cup Software Design Competition
2013	55000USD Project Grant from Turkish Ministry of Science
2013	National University of Singapore best entrepreneurship idea award
2014	NWO, PhD Scholarship top-grant award

LIST OF PUBLICATIONS

- Sem2Vec: Semantic Word Vectors with Bidirectional Constraint Propagations
  Taygun Kekeç, David M. J. Tax
  IEEE TKDE: Transactions on Knowledge and Data Engineering (under review).

- Boosted Negative Sampling by Quadratically Constrained Entropy Maximization
  Taygun Kekeç, David Mimno, David M. J. Tax
  PRL: Pattern Recognition Letters.

- PAWE: Polysemy Aware Word Embeddings
  Taygun Kekeç, Laurens van der Maaten, David M. J. Tax
  ICISDM: International Conference on Information Systems and Data Mining , 2018.

- Supervising topic models with Gaussian processes
  Melih Kandemir, Taygun Kekeç, Reyyan Yeniterzi
  PR: Pattern Recognition, 2018.

- Markov Random Field for Wind Farm Planning
  Hâle Çetinay, Taygun Kekeç, Fernando A. Kuipers, David M. J. Tax
  SEGE: International Conference on Smart Energy Grid Engineering, 2017.

- Robust Gram Embeddings
  Taygun Kekeç, David M. J. Tax
  EMNLP: Empirical Methods in Natural Language Processing, 2016.

- Contextually Constrained Deep Networks for Scene Labeling
  Taygun Kekeç, Rémi Emonet, Elisa Fromont, Alain Trémeau, Christian Wolf
  BMVC: British Machine Vision Conference, 2014.

- Prise en Compte du Contexte pour Contraindre les Réseaux Profonds
  Taygun Kekeç, Rémi Emonet, Elisa Fromont, Alain Trémeau, Christian Wolf
  CAP: Conférence d'Apprentissage Automatique, 2014.

- A New Approach to Real-time Mosaicing of Aerial Images
  Taygun Kekeç, Alper Yıldırım, Mustafa Ünel
  Robotics and Autonomous Systems, 2014.

- A Modular Software Architecture for UAVs
  Taygun Kekeç, Baris Can Ustundag, M. A. Guney, Alper Yıldırım, Mustafa Ünel.
  IECON: 39th Annual Conference of the IEEE Industrial Electronics Society, 2013.

- Real-time Multiple Object Tracking using Virtual Shells
  Taygun Kekeç, Mustafa Ünel, Hakan Erdoğan
  ICMCA: International Conference on Mathematical and Computational Applications, 2013.

- Insansiz Hava Araclari icin Donanimdan Bagimsiz Yazilim Sistemi Gelistirilmesi
  Baris Can Ustundag, Taygun Kekeç, M. A. Guney, Pamir Mundt, Mustafa Ünel
  TOK: Otomatik Kontrol Ulusal Toplantisi, 2013.

- Insansiz Hava Araclari icin Moduler bir Sistem Mimarisi
  Taygun Kekeç, Baris Can Ustundag, Mustafa Ünel
  TOK: Otomatik Kontrol Ulusal Toplantisi, 2012.