國立臺灣大學電機資訊學院資訊工程學研究所
碩士論文
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

非監督式語意表徵的強化學習方法
Unsupervised Sense Representation by Reinforcement Learning

李廣和
Guang-He Lee

指導教授：陳縕儂　博士
Advisor: Yun-Nung Chen, Ph.D.

中華民國 106 年 6 月
June, 2017

doi:10.6342/NTU201701669

國立臺灣大學碩士學位論文
口試委員會審定書

非監督式語意表徵的強化學習方法

Unsupervised Sense Representation by Reinforcement Learning

本論文係李廣和君（學號 R04922045）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 106 年 6 月 23 日承下列考試委員審查通過及口試及格，特此證明

口試委員：陳縕儂

（指導教授）

林軒田

張訓嘉

系主任 趙坤茂

摘要

本論文嘗試以非監督式的方法解決語意混淆問題，其語意表徵的學習必須建立在有情景的語意選擇功能之下。過往在學習語意表徵的研究多半無法兼顧表徵學習的精細度與語意選擇的效率性。本論文提出了一個模組化的架構，支持靈活的模組來優化各自的目標：一模組選擇詞對應之語意，另一模組針對選到的語意來學習表徵向量，因此達成第一個支持線性語意選擇功能的純語意階層表徵學習模型。對比於傳統的架構，我們使用了加強學習來達成了以下三種好處。一、加強學習的決策架構比起機率與分群更能描述人在選擇語意的機制；二、我們藉由加強學習的方法，提出了在模組化架構底下第一個能只用單一目標函數的非監督式語意表徵模型；三、我們更在語意選擇中引入了加強學習中多元的探索功能來增加穩健性。在基準資料集的實驗結果顯示出本論文的方法在同義字選擇與 (最高餘弦相似度的) 情景字相似性的實驗中都超越了最先進的方法。

關鍵字：非監督式語意表徵；表徵學習；加強學習。

doi:10.6342/NTU201701669

Abstract

This paper proposes to address the word sense ambiguity issue in an unsupervised manner, where word sense representations are learned along a word sense selection mechanism given contexts. Prior work about learning sense embeddings suffered from either coarse-grained representation learning or inefficient sense selection. The proposed modular framework implements flexible modules to optimize distinct mechanisms: sense selection and representation learning, achieving the first purely sense-level representation learning system with linear-time sense selection. In contrast to conventional methods, we leverage reinforcement learning as the learning algorithm, which exhibits the following advantages. First, the decision making process under reinforcement learning better captures the sense selection mechanism than probabilistic and clustering methods. Second, our reinforcement learning algorithm realizes the first single objective function for modular unsupervised sense representation systems. Finally, we introduce various exploration techniques under reinforcement learning on sense selection to enhance robustness. The experiments on benchmark data show that the proposed approach achieves the state-of-the-art performance on synonym selection as well as on contextual word similarities in terms of MaxSimC.

Keywords. Unsupervised Sense Representation; Representation Learning; Reinforcement Learning.

doi:10.6342/NTU201701669

Contents

doi:10.6342/NTU201701669

doi:10.6342/NTU201701669

List of Figures

doi:10.6342/NTU201701669

List of Tables

doi:10.6342/NTU201701669

Chapter 1

Introduction

1.1 Motivation

Natural language processing (NLP) is one of the biggest challenge for artificial intelligence (AI). Compared to other sensory mechanism like vision or sounds, language is artificially coined to carry semantics and such surface form of semantics, language, is highly ambiguous: even the smallest semantic unit in language, word, is polysemous. That says, repetitively using the same word to carry different meanings is a frequent phenomenon for modern natural language. For example, the word "apple" may either refer to a kind of fruit or a computer brand.

To resolve the rudimentary problem in NLP, we try to decipher the hidden semantics of words: disentangling different semantics of word to different semantic representations. In addition, since no sufficient annotation is available for this task, we aim to do this task in an unsupervised manner. Specifically, we try to leverage big language data, i.e., corpus, to automatically discover the pattern of occurrence of distinct word senses, and build corresponding representation to describe its semantics.

1.2 Word Representation

The development of NLP algorithms highly depends on the semantic representation of input language data, since a prediction model cannot be built without such interface to feed data into machine learning algorithms. Conventionally, such interface or data representation for NLP highly relied on hand-crafted feature to represent a word, sentence, or

doi:10.6342/NTU201701669

paragraph. However, such method has its limitation since development of useful feature on NLP typically requires the knowledge from linguistic experts and thus demands high cost.

To circumvent such manual development of data representation for NLP, researchers also proposed to treat each word token as a discrete symbol in a set or an one-hot vector v. For example, given a vocabulary as {"I", "want", "to", "eat", "apple", "banana", "orange"}, the word "apple" can be represented as an one-hot vector as (1.1).

$$[0, 0, 0, 0, 1, 0, 0]. \tag{1.1}$$

Accordingly, similar technique is applied to sentence or paragraph that treats data as bag-of-words (BOW) in a vector. For example, the sentence "I want to eat banana" can be represented as (1.2)

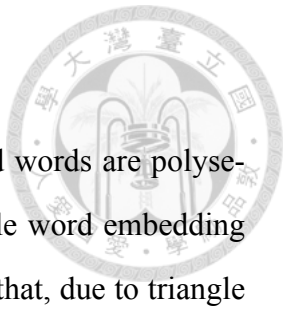
$$[1, 1, 1, 1, 0, 1, 0]. \tag{1.2}$$

Despite simple data representation is available by using one-hot vectors or BOW, such representation has a significant drawback that does not take the relation between words into account. For example, the word "banana" is semantically more similar to "apple" than "want", but the above representation demonstrates orthogonal relationship among words equally, neglecting the nuances of semantic relation among words.

To address the above problem, continuous representation is proposed to NLP, that embeds subtle relation between data in a more expressive continuous space than discrete space. The framework includes lots of models that may either learn the representation for documents [2, 3] or words [4, 5], where the word representation is also called word embedding. The method can demonstrate several linguistic relationship in the embedding space, such as female-male and country-capital relation [4]. Most importantly, such phenomenon is achieved by learning from a corpus without manual annotations.

doi:10.6342/NTU201701669

1.3 Sense Representation



However, considering that natural language is highly ambiguous and words are polysemous, representing possibly manifold semantics of a word in a single word embedding may suffer from polysemy issues. Neelakantan et al. [6] pointed out that, due to triangle inequality in vector space, if one word has two different senses but is restricted to *one embedding*, the sum of the distances between the word and its synonym in each sense would upper-bound the distance between the respective synonyms, which may be mutually irrelevant, in embedding space. For example, "stone" and "shake" are irrelevant, but due to the mutual synonym (in different senses), "rock", the embedding distance between "stone" and "shake" is bounded by the embedding distance from their mutual synonym due to triangle inequality, $d(\mathrm{rock}, \mathrm{stone}) + d(\mathrm{rock}, \mathrm{shake}) \geq d(\mathrm{stone}, \mathrm{shake})$.

Due to the theoretical inability to account for polysemy using a single embedding representation per word, multi-sense word representations are proposed to address the ambiguity issue using multiple embedding representations for different senses in a word [7, 8].

This thesis focuses on unsupervised learning for sense representation from an unannotated corpus. Specifically, we follow the convention to embed semantics into a continuous space, i.e., distributed representations [4]. There are two key mechanisms for a multi-sense word representation system in such scenario:

1. an unsupervised sense selection (decoding) mechanism infers the most probable sense for a word given its context, and
2. a sense representation mechanism learns to embed word senses in a continuous space.

Under this framework, prior work focused on designing a single model to deliver both mechanisms [6, 9, 10]. However, the previously proposed models introduce side-effects:

1. mixing word-level and sense-level tokens achieves efficient sense selection but introduces ambiguous word-level tokens during the representation learning process [6, 9], and
2. pure sense-level tokens prevent ambiguity from word-level tokens but require ex-

doi:10.6342/NTU201701669

ponential time complexity when decoding a sense sequence [10].

1.4 Contributions

Unlike prior work, this thesis proposes *MUSE*—Modularizing Unsupervised Sense Embeddings—a novel modularization framework incorporating sense selection and representation learning models, which implements flexible modules to optimize distinct mechanisms. Specifically, MUSE enables linear time sense identity decoding with a *sense selection module* and purely sense-level representation learning with a *sense representation module*.

With the modular design, we propose a novel joint learning algorithm on the modules by connecting to a reinforcement learning scenario, which achieves the following advantages. First, the decision making process under reinforcement learning better captures the sense selection process than conventional frameworks using probabilistic or clustering methods. Second, our reinforcement learning algorithm achieves the first modular unsupervised sense representations system with a single objective. Finally, we introduce various exploration techniques under reinforcement learning on sense selection to enhance robustness.

Furthermore, since no reliable supervision signal is available, we investigate theoretical drawback of the proposed framework, and propose a stable learning algorithm. In summary, our contributions are five-fold:

- *MUSE* is the first system that maintains purely sense-level representation learning with linear-time sense decoding.
- We are among the first to leverage reinforcement learning to model the decision making process for sense selection in sense representations system.
- We are among the first to propose a single objective for modularized unsupervised sense embedding learning.
- We introduce a sense exploration mechanism for the sense selection module to achieve better flexibility and robustness.
- Our experimental results show the state-of-the-art performance for synonym selec-

doi:10.6342/NTU201701669

tion and contextual word similarities in terms of MaxSimC.

Part of this research work has been presented in the publication [11]: Guang-He Lee and Yun-Nung Chen "MUSE: Modulizing Unsupervised Sense Embeddings," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, Copenhagen, Denmark, 2017.

1.5 Thesis Structure

In the following chapters, we will first review technical backgrounds for the proposed methods in the Chapter 2, and elaborate the proposed methods in the Chapter 3. Afterwards, both qualitative experiments and quantitative experiments against state-of-the-art competitors are available in the Chapter 4. Then Chapter 5 describes the details of the competitors. Finally, the Chapter 6 conclude this work.

doi:10.6342/NTU201701669

Chapter 2

Background Review

In this chapter, we review technical backgrounds that are used for developing the proposed methods.

2.1 Word Representation Learning

The word representation learning models embed each word token into a continuous space (i.e., vector space). Such semantic representation is typically learned through maximum likelihood estimation (MLE) for collocation likelihood. Here we review some popular representation learning algorithms

The input word representation matrix U and collocation estimation matrix V are first established as learning targets. To estimate collocation likelihood, a natural choice is to formulate a categorical distribution over all possible words given the target word w_i and collocated word w_j using a softmax function:

$$\log \mathcal{L}(w_j \mid w_i) = \log \frac{\exp(U_{w_i}^T V_{w_j})}{\sum_{w_k} \exp(U_{w_i}^T V_{w_k})}. \tag{2.1}$$

Instead of enumerating all possible collocated words which is computationally expensive, the skip-gram model [4] is proposed to approximate (2.1) as (2.2).

$$\log \bar{\mathcal{L}}(w_j \mid w_i) = \log \sigma(U_{w_i}^T V_{w_j}) + \sum_{c=1}^{M} \mathbb{E}_{w_k \sim p_{neg}(w)}[\log \sigma(-U_{w_i}^T V_{w_k})], \tag{2.2}$$

where $p_{neg}(w)$ is some distribution over all words for negative samples. The concept of skip-gram model is to approximate the softmax function that contrasts the collocation with

doi:10.6342/NTU201701669

all other tokens as a logistic regression that contrasts the collocation with randomly drawn negative tokens. As a similar idea, the CBOW objective [12] is proposed to extend the concept of collocation to a group of collocated tokens. Denoting $\bar{C}_t$ as the set of words in the local context , CBOW is defined as (2.3).

$$\log \bar{\mathcal{L}}(w_i \mid \tilde{C}_t) = \log \sigma(\sum_{w_j \in \bar{C}_t} U_{w_j}^T V_{w_i}) + \sum_{c=1}^{M} \mathbb{E}_{w_k \sim p_{neg}(w)}[\log \sigma(-\sum_{w_j \in \bar{C}_t} U_{w_j}^T V_{w_k})], \quad (2.3)$$

Finally, GloVe [5] is proposed to explicitly incorporate counting statisitcs in the objective function[1]. Denoting $X_{w_i w_j}$ as the cooccurence counts for token w_i and w_j across corpus, GloVe learns the representation by minimizing the square distance between log coocurence counts and prediction as

$$\min_{U,V} \quad f(X_{w_i w_j})(\log X_{w_i w_j} - U_{w_i}^T V_{w_j})^2, \quad (2.4)$$

where $f(X_{w_i w_j})$ is a pre-defined function that weights collocation w_i and w_j. As a weighted square error is adopted by GloVe, it can also be regarded as maximization of a heteroscedastic Gaussian likelihood, where $f(X_{w_i w_j})$ associates with the variance and (collocation) cooccurence counts associates with the mean, so GloVe can also be loosely regarded as a collocation likelihood maximization scenario.

2.2 Deep Learning for Natural Language Processing

Most of the tasks in NLP can be casted as a machine learning task that predicts a score given a context. To build such inference model, deep learning is arguably the dominant choice, which includes Deep Neural Networks (DNN) [13], Convolutional Neural Networks (CNN) [14], and Recurrent Neural Networks (RNN) [15]. Each above model use different modeling method to learn an inference model $f(\cdot)$ from data, which introduces different pros and cons on respective modeling method. First, DNN iteratively use a matrix product with a (non-linear) activation function on each data point to conduct inference, which is not able to distinguish contextual information since it omits both the temporal and

[1] All models that directly model collocation likelihood in a corpus (e.g., skip-gram) also implicitly incorporates cooccurence counts, since the frequency of updates of each pair of collocation in an epoch is equivalent to cooccurence counts.

doi:10.6342/NTU201701669

local information. Second, CNN *locally* conducts a matrix product with a (non-linear) activation function to conduct inference, which extracts local information but may not extract rich temporal information unless a deep model architecture. Finally, RNN conducts local matrix products with both current input and *temporal* input, which extracts temporal information but involves much more complex model architecture than the previous methods.

2.3 Markov Decision Process and Reinforcement Learning

2.3.1 Markov Decision Process

An Markov Decision Process (MDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ denotes a set of states, $\mathcal{A}$ denotes a set of actions, $\mathcal{P}$ denotes a stochastic transition matrix for state transition $\mathcal{P}^a_{ij} = \mathbb{P}(S_{t+1} = j \mid S_t = i, A_t)$ ($S_t, S_{t+1} \in \mathcal{S}, A_t \in \mathcal{A}$), $\mathcal{R}$ denotes a reward function $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$ (R_{t+1} is a random variable denoting the reward at timestamp $t+1$), and $\gamma \in [0, 1]$ is a discount factor.

Given a starting state S_0 and a policy distribution (2.5),

$$\pi'(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s], \tag{2.5}$$

we can sample a trajectory of states $(S_t, \cdots)$, actions $(A_t, \cdots)$, and corresponding rewards $(R_t, \cdots)$ from time t, and compute the accumulated reward (also known as return) as

$$G_t = \sum_{i=0}^{\infty} \gamma^i R_{t+i+1}, \tag{2.6}$$

where the state transition from S_t to S_{t+1} distributed from the state transition matrix $\mathcal{P}$ with an action a distributed from a policy distribution (2.5).

With a policy distribution π', we can further define the action value function $q^*(a \mid s)$, which is also known as Q-value, for the policy as the expected accumulated reward from a starting state s and a following action a:

$$q^*(a \mid s) = \mathbb{E}_{\pi'}[G_t \mid S_t = s, A_t = a]. \tag{2.7}$$

doi:10.6342/NTU201701669

2.3.2 Reinforcement Learning

The literature of reinforcement learning can be divided to the following taxonomy. First, a policy-based reinforcement learning algorithm tries to maximize (2.7) by learning the probabilistic policy $\pi'(a \mid s)$. Second, a value-based reinforcement learning algorithm directly estimates state value or action value $q^*(a \mid s)$ as $q'(a \mid s)$ given a deterministic policy $\pi'(a \mid s)$ as (2.8).

$$\pi'(a \mid s) = \begin{cases} 1, & \text{if } a = \arg\max_{a'} q'(a' \mid s) \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

Third, an actor-critic based reinforcement learning algorithm combines both policy-based and value-based methods.

Even though an optimal policy can be obtained by table filling, the policy distribution table is typically too large (Cartesian product between state space and action space) to fit in memory. Practically, researchers use reinforcement learning by using function approximators like neural networks to learn a probabilistic policy $\pi_\theta(a \mid s)$ or an action value $q_\theta(a \mid s)$ with parameter set θ [16].

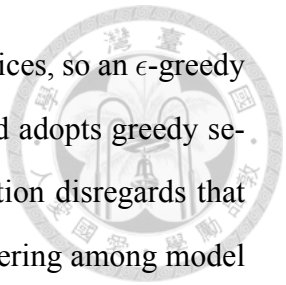
2.3.3 Exploitation and Exploration

Despite the fact that reinforcement learning provides learning algorithm for MDP, it still requires an important mechanism to obtain data for model training. That says, a trajectory of states $(S_t, \cdots)$, actions $(A_t, \cdots)$, and corresponding rewards $(R_t, \cdots)$ must be first obtained before model training.

However, the trajectory of actions come from a series of decision making based on estimated action value or policy from the model. Such decision making reveals a dilemma that a model should exploit its best estimation if the estimations are trustworthy, or explore underestimated actions otherwise. Since most of the models do not generates uncertainty estimation for each action, the choice between exploitation and exploration becomes non-trivial.

Under this scenario, the most basic idea, greedy selection, is to always exploit the best

doi:10.6342/NTU201701669



estimation. Nevertheless, such method disregards underestimated choices, so an ϵ-greedy selection, which uniformly explores all actions with ϵ probability and adopts greedy selection otherwise, may be a better choice. However, ϵ-greedy selection disregards that despite inaccurate exact values on model estimations, the relative ordering among model estimations may still be correct. Accordingly, an even better selection strategy, Boltzmann sampling, is to sample actions weighted by model estimations, which essentially normalizes model estimations to a probability distribution.

doi:10.6342/NTU201701669

Chapter 3

Methodology

This work proposes a framework to modularize two key mechanisms for multi-sense word representations: a *sense selection module* and a *sense representation module*. The sense selection module decides which sense to use given a text context, whereas the sense representation module learns meaningful representations based on its statistical characteristics. Unlike prior work that must compromise between efficient sense selection and purely sense-level representation learning, the proposed modularized framework is capable of performing efficient sense selection and learning representations in pure sense level simultaneously.

To learn sense-level representations, a sense selection model should be first established for sense identity decoding. On the other hand, the sense embeddings should guide the sense selection model when decoding a sense identity sequence. Therefore, these two modules should be tangled. This indicates that a naive two-stage algorithm or two separate learning algorithms proposed by prior work are not optimal.

By connecting the proposed formulation with reinforcement learning literature, we design a novel joint training algorithm. In addition, we prove the theoretical limitation of some reinforcement learning algorithms in this unsupervised task, and further propose several stabilization techniques. Besides, taking advantage of the form of reinforcement learning, we are among the first to investigate various exploration techniques in sense selection for unsupervised sense embedding learning.

doi:10.6342/NTU201701669

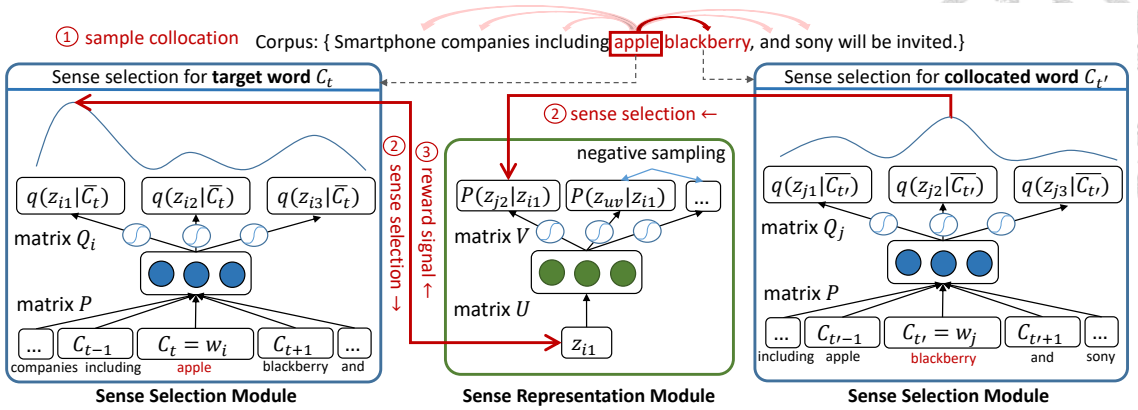


Figure 3.1: The *MUSE* architecture with a 3-step learning algorithm: 1) collocation sampling, 2) sense selection for sense representation learning, and 3) optimizing sense selection with a reward signal from sense representation. Reward signal is only passed to the target word to stabilize model training due to directional architecture in the sense representation module.

3.1 Model Architecture

Our model architecture is illustrated in Figure 3.1, where there are two modules in optimization.

3.1.1 Sense Selection Module

Generally speaking, to perform a selection task over possible candidates Y given an input x, we should first build a model that infers a score $\bar{y}$ for each possible candidate $y \in Y$. Despite the variety of manifold modeling methods such as discriminative model, generative model [17], and regression model [18], it is universal to associate each candidate with a score representing the respective fitness.

In our task, given a corpus C, vocabulary V, and the t-th word $w_i = C_t \in V$, we would like to find the most probable sense $z_{ik} \in Z_i$, where Z_i is the set of senses in word w_i. Assuming that a word sense is determined by the local context, we exploit a local context $\bar{C}_t = \{C_{t-m}, \cdots, C_{t+m}\}$ for sense selection according to the Markov assumption, where m is the size of a context window. Then we can either formulate a probabilistic policy $\pi(z_{ik} \mid \bar{C}_t)$ about sense selection or estimate the *individual* fitness $q(z_{ik} \mid \bar{C}_t)$ for each sense identity.

For unsupervised sense representation learning, efficiency is arguably the most critical

doi:10.6342/NTU201701669

factor affecting the performance since each training epoch may involve up to 120 billion word tokens for decoding in the training corpus [9], when the sense selection step may be treated as an additional overhead for sense representation learning compared to word representation learning. Hence, we choose DNN as the inference model for sense selection. Here we refer the readers to [13, 14, 15] for details about other neural models on related NLP task, and will only provide the technical details for the DNN model that we will adopt.

To achieve efficiency, in this work we exploit a linear DNN architecture that takes word-level input tokens and outputs sense-level identities. The architecture is similar to continuous bag-of-words (CBOW) [12]. Specifically, given a *word* embedding matrix P, the local context can be modeled as the summation of word embeddings from its context $\bar{C}_t$. The output can be formulated with a 3-mode tensor Q, whose dimensions denote words, senses, and latent variables. Then we can model $\pi(z_{ik} \mid \bar{C}_t)$ or $q(z_{ik} \mid \bar{C}_t)$ correspondingly. Here we model $\pi(\cdot)$ as a *categorical* distribution using a softmax layer:

$$\pi(z_{ik} \mid \bar{C}_t) = \frac{\exp(Q_{ik}^T \sum_{j \in \bar{C}_t} P_j)}{\sum_{k' \in Z_i} \exp(Q_{ik'}^T \sum_{j \in \bar{C}_t} P_j)}. \tag{3.1}$$

On the other hand, the likelihood of selecting *distinct* sense identities, $q(z_{ik} \mid \bar{C}_t)$, is modeled as a *Bernoulli* distribution with a sigmoid function $\sigma(\cdot)$:

$$q(z_{ik} \mid \bar{C}_t) = \sigma(Q_{ik}^T \sum_{j \in \bar{C}_t} P_j). \tag{3.2}$$

Different modeling approaches lead to different learning methods, especially for this unsupervised setting. Here we only list possible modeling settings for the sense selection module and leave the learning algorithm on § 3.2 when the sense representation module is also presented. Finally, with a built sense selection module, we can apply a selection algorithm such as greedy selection strategy to infer the sense identity z_{ik} given a word w_i with its context C_t.

Finally, regardless of model architecture, we note that modularized model allows sense selection to enjoy efficiency by leveraging word-level tokens, while remaining purely sense-level tokens in the representation module. Specifically, if n denotes $\max_k |Z_k|$, de-

doi:10.6342/NTU201701669

coding L words takes $O(nL)$ senses to be searched due to independent sense selection. The prior work using a single model with purely sense-level tokens [10] requires exponential time to calculate the collocation energy for every possible combination of sense identities within a context window, $O(n^{2m})$, given a *single target sense*. Further, the prior work [10] took an additional sequence decoding step with quadratic time complexity $O(n^{4m}L)$, based on an exponential number n^{2m} in the base unit. It demonstrates the achievement of our proposed model on efficient sense inference.

3.1.2 Sense Representation Module

Once a sense selection mechanism is established, the main concern for unsupervised sense representation learning is to learn the semantic representation of sense from data. While similar techniques as word representation learning is directly applicable to sense representation learning, sense representation learning is applied to sense tokens, which should be first decoded by a sense selection mechanism.

Similar to word representation models, we first create input sense representation matrix U and collocation estimation matrix V as the learning targets. Given a target word w_i and collocated word w_j with corresponding local contexts, we map them to their sense identities as z_{ik} and z_{jl} by the sense selection module, and maximize the sense collocation log likelihood $\log \mathcal{L}(\cdot)$.

To select representation learning model for unsupervised sense representation learning, efficiency and scalability is far more critical than word representation learning for the following three reasons. First, a light-weight representation learning model is prefered for scalability since the sense-level vocabulary must be larger than the word-level vocabulary, assuming no sense token belongs to multiple word tokens. Second, since both modules may be unstable without supervised signal, stochastic optimization that only affects a few instance is more likely to escape poor local optimum from randomly inadequate sense selection than batch optimization. Finally and most importantly, sense selection introduces overhead for representation learning, so it is desirable for the representation learning model to depend on only a few instances.

doi:10.6342/NTU201701669

Hence, we use the skip-gram formulation [4] that only takes two sense identities for stochastic training as in (3.3) (word level version equation is in (2.2)), whereas CBOW [12] requires sense selection for the whole context window as in (2.3) and GloVe [5] takes computationally expensive collocation *counting* statistics for each token in a corpus for each optimization step as in (2.4).

$$\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) = \log \sigma(U_{z_{ik}}^T V_{z_{jl}}) + \sum_{c=1}^{M} \mathbb{E}_{z_{uv} \sim p_{neg}(z)}[\log \sigma(-U_{z_{ik}}^T V_{z_{uv}})], \tag{3.3}$$

Finally, We note that our modular framework can easily maintain purely sense-level tokens with an arbitrary representation learning model. In contrast, most related work using probabilistic modeling [19, 20, 9, 21] binded sense representations with the sense selection mechanism, so efficient sense selection by leveraging word-level tokens can be achieved only at the cost of mixing word-level and sense-level tokens in their representation learning process.

3.2 Joint Formulation

Without supervised signal for the proposed modules, it is desirable to connect two modules in a way where they can improve each other by their own estimations. On one hand, forwarding the prediction of the sense selection module to the representation module is trivial. On the other hand, we cast the estimated collocation likelihood as a reward signal for the selected sense for effective learning. We cast the whole process as an MDP, and solve the optimization problem by reinforcement learning [22]. In addition, based on different modeling methods ((3.1) or (3.2)) in the sense selection module, we connect the model to respective reinforcement learning algorithms.

3.2.1 Markov Decision Process

We proposed that unsupervised sense representation learning can be treated as a two-step MDP that involving a target word and a collocation, where the state S_t, action A_t, and accumulated reward G_t correspond to context $\bar{C}_t$, sense z_{ik}, and collocation log likeli-

doi:10.6342/NTU201701669

hood $\log \bar{\mathcal{L}}(\cdot)$ respectively. Besides, we treat the state transition matrix $\mathcal{P}$ as a uniformly stochastic transition from the target word to a collocation in the local context $\bar{C}_t$, and set the discount factor γ as 1.

The proposed MDP framework embodies several nuances of sense selection. First, the decision of a word sense is Markov: taking the whole corpus into consideration is not more helpful than a handful of necessary local contexts. Second, the decision making in MDP exploits a hard decision for selecting sense identity, which captures the sense selection process more naturally than forming a probabilistic distribution over possible sense identities. Finally, we exploits the reward mechanism in MDP to enable joint training: the estimation of sense representation is treated as a reward signal to guide sense selection. In contrast, the decision making under clustering considers the similarity within clusters, rather than the outcome of a decision using a reward signal as MDP.

3.2.2 Reinforcement Learning

To connect our method to reinforcement learning, we refer $\pi(z_{ik} \mid \bar{C}_t)$ in (3.1) to estimated policy distribution $\pi_\theta(a \mid s)$ and refer $q(z_{ik} \mid \bar{C}_t)$ in (3.2) to estimated Q-value $q_\theta(a \mid s)$ in the reinforcement learning literature. Finally, note that our following derivation focuses on stochastic optimization for a practical learning setting.

Policy Gradient Method

Because (3.1) fits a valid probability distribution, an intuitive way to perform learning is to optimize the expectation of resulting collocation likelihood among each sense. That says, an optimal policy $\pi(\cdot)$ in (3.1) should maximize the collocation likelihood in expectation (3.4).

$$\mathbb{E}_{z_{ik} \sim \pi(\cdot \mid \bar{C}_t)}[\mathbb{E}_{z_{jl} \sim \pi(\cdot \mid \bar{C}_{t'})}[\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik})]], \tag{3.4}$$

where $\pi(\cdot \mid \bar{C}_t)$ and $\pi(\cdot \mid \bar{C}_{t'})$ stand for the probabilistic policy for selecting senses of word $w_i = C_t, w_j = C_{t'}$ given context $\bar{C}_t, \bar{C}_{t'}$. The objective is differentiable and supports stochastic optimization [23], which uses a stochastic samples z_{ik} and z_{jl} for optimization.

However, there are two possible disadvantages in this formulation. First, because the

doi:10.6342/NTU201701669

policy assumes the probability distribution in (3.1), optimizing the selected sense must affect the estimation of the other senses. Second, if applying stochastic gradient ascent to optimizing (3.4), it would always lower the probability estimation for the selected senses z_{ik} and z_{jl} even if the model accurately selects the right senses.

To prove the second disadvantage, we derive doubly stochastic gradient for equation (3.4). We first denote (3.4) as $J(\Theta)$ with $\Theta = \{P, Q\}$ and resolve the expectation form as:

$$\begin{aligned} J(\Theta) &= \mathbb{E}_{z_{ik}\sim\pi(\cdot|\bar{C}_t)}[\mathbb{E}_{z_{jl}\sim\pi(\cdot|\bar{C}_{t'})}[\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik})]] \\ &= \sum_k \sum_l \pi(z_{ik} \mid \bar{C}_t)\pi(z_{jl} \mid \bar{C}_{t'}) \log \bar{\mathcal{L}}(z_{jl}|z_{ik}). \end{aligned} \tag{3.5}$$

The gradient with respect to Θ should be:

$$\begin{aligned} \frac{\partial J(\Theta)}{\partial \Theta} &= \frac{\partial}{\partial \Theta} \sum_k \sum_l \pi(z_{ik} \mid \bar{C}_t)\pi(z_{jl} \mid \bar{C}_{t'}) \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) \\ &= \sum_k \sum_l \pi(z_{ik} \mid \bar{C}_t) \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) \frac{\partial}{\partial \Theta} \pi(z_{jl} \mid \bar{C}_{t'}) \\ &+ \sum_k \sum_l \pi(z_{jl} \mid \bar{C}_{t'}) \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) \frac{\partial}{\partial \Theta} \pi(z_{ik} \mid \bar{C}_t) \\ &= \sum_k \sum_l \pi(z_{ik} \mid \bar{C}_t) \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) \pi(z_{jl} \mid \bar{C}_{t'}) \frac{\partial}{\partial \Theta} \log \pi(z_{jl} \mid \bar{C}_{t'}) \\ &+ \sum_k \sum_l \pi(z_{jl} \mid \bar{C}_{t'}) \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) \pi(z_{ik} \mid \bar{C}_t) \frac{\partial}{\partial \Theta} \log \pi(z_{ik} \mid \bar{C}_t) \\ &= \mathbb{E}_{z_{ik}\sim\pi(\cdot|\bar{C}_t)}[\mathbb{E}_{z_{jl}\sim\pi(\cdot|\bar{C}_{t'}}[\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik})(\frac{\partial}{\partial \Theta} \log \pi(z_{ik} \mid \bar{C}_t) + \frac{\partial}{\partial \Theta} \log \pi(z_{jl} \mid \bar{C}_{t'}))]]. \end{aligned} \tag{3.6}$$

Accordingly, if we conduct typical stochastic gradient ascent training on $J(\Theta)$ with respect to Θ from samples z_{ik} and z_{jl} with a learning rate η, the update formula will be:

$$\Theta = \Theta + \eta \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik})(\frac{\partial}{\partial \Theta} \log \pi(z_{ik} \mid \bar{C}_t) + \frac{\partial}{\partial \Theta} \log \pi(z_{jl} \mid \bar{C}_{t'})). \tag{3.7}$$

However, the collocation log likelihood should always be non-positive: $\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) \leq 0$. Therefore, as long as the collocation log likelihood $\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik})$ is negative, the update formula is to minimize the likelihood of choosing z_{ik}, despite the fact that z_{ik} may be a good choice. On the other hand, if the log likelihood reaches 0, according to (3.3), it

doi:10.6342/NTU201701669

indicates:

$$\begin{aligned} & \log \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) = 0 \\ \Leftrightarrow \quad & \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) = 1 \\ \Leftrightarrow \quad & U_{z_{ik}}^T V_{z_{jl}} \to \infty, \;\; U_{z_{ik}}^T V_{z_{uv}} \to \infty, \;\; \forall z_{uv}, \end{aligned} \tag{3.8}$$

which leads to computational overflow from an infinity value.

To avoid the adversarial effect of negative reward for policy gradient, we propose another variant that transform the reward signal to the range between $[0, 1]$. Specifically, we replace the log likelihood $\log \bar{\mathcal{L}}(\cdot) \in (-\infty, 0]$ with the likelihood $\bar{\mathcal{L}}(\cdot) \in [0, 1]$ as the reward function. Due to the monotonic operation in $\log(\cdot)$, the relative ordering of the reward remains the same. We note that reducing the reward range to aid model training [16] is a common practice to stabilize model training for neural reinforcement learning. We denote the original version as MUSE-Policy and the smoothed formulation as MUSE-Smooth-Policy.

Value-Based Method

To address the above issues, we exploit the Q-learning algorithm [16]. Instead of maintaining a probabilistic policy for sense selection, Q-learning estimates the Q-value (resulting collocation log likelihood in expectation) for each sense candidate directly and independently. Thus, the estimation of unselected senses may not be influenced by the selected one. Note that in MDP, the (expected accumulated) reward $\mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$ is equivalent to Q-value $q^*(a \mid s)$, so we will use reward and Q-value interchangeably, hereinafter, based on the context.

Since Q-learning estimates value function instead of a probabilistic policy, the reward function smoothing can be obtained in a more principled way than policy gradient method. To elaborate, since the reward function is log likelihood, we can impose an additional $\log(\cdot)$ activation function to the Bernoulli Q-value prediction in (3.2), which also results in a log likelihood value. Hence, the reward value and Q-value will lie in the same numerical space. Equivalently, we can transform the reward function to probability space to achieve

doi:10.6342/NTU201701669

the same effect.

In addition, we can exploit the probabilistic nature of likelihood for Q-learning. To elaborate, as Q-learning is used to approximate the Q-value for each action in typical reinforcement learning scenario, most literature adopted square loss to characterize the discrepancy between the target and estimated Q-values [16]. In our setting where the Q-value/reward is a likelihood function, our model exploits cross entropy[1] to better capture the characteristics of probability distribution.

Furthermore, given that the collocation likelihood in (3.3) is an *approximation* to the original categorical distribution in (2.1) [4], we revise the formulation by omitting the negative sampling term. The resulting formulation $\hat{\mathcal{L}}(\cdot)$ is a Bernoulli distribution indicating whether z_{jl} collocates or not given z_{ik}:

$$\hat{\mathcal{L}}(z_{jl} \mid z_{ik}) = \sigma(U_{z_{ik}}^T V_{z_{jl}}). \tag{3.9}$$

There are three advantages of using $\hat{\mathcal{L}}(\cdot)$. First, regarding the variance of estimation, $\hat{\mathcal{L}}(\cdot)$ better captures $\mathcal{L}(\cdot)$ than $\bar{\mathcal{L}}(\cdot)$ because $\bar{\mathcal{L}}(\cdot)$ involves sampling:

$$Var(\bar{\mathcal{L}}(\cdot)) \geq Var(\hat{\mathcal{L}}(\cdot)) = Var(\mathcal{L}(\cdot)) = 0. \tag{3.10}$$

Second, regarding the relative ordering of estimations, for any two collocated senses z_{jl} and $z_{jl'}$ with a target sense z_{ik}, $\hat{\mathcal{L}}(\cdot)$ is equivalent to $\mathcal{L}(\cdot)$:

$$\begin{aligned} & \mathcal{L}(z_{jl} \mid z_{ik}) < \mathcal{L}(z_{jl'} \mid z_{ik}) \\ \Leftrightarrow \quad & \bar{\mathcal{L}}(z_{jl} \mid z_{ik}) < \bar{\mathcal{L}}(z_{jl'} \mid z_{ik}) \\ \Leftrightarrow \quad & \hat{\mathcal{L}}(z_{jl} \mid z_{ik}) < \hat{\mathcal{L}}(z_{jl'} \mid z_{ik}) \end{aligned} \tag{3.11}$$

Third, for collocation computation, $\mathcal{L}(\cdot)$ requires all sense identities and $\bar{\mathcal{L}}(\cdot)$ requires $(M+1)$ sense identities, whereas $\hat{\mathcal{L}}(\cdot)$ requires only 1 sense identity. In sum, the proposed $\hat{\mathcal{L}}(\cdot)$ approximates $\mathcal{L}(\cdot)$ with no variance, no "bias" (in terms of relative ordering), and significantly less computation.

Finally, because both target distribution $\hat{\mathcal{L}}(\cdot)$ and estimated distribution $q(\cdot)$ are Bernoulli

[1] In this case where the reward is fixed, cross entropy is equivalent to KL divergence.

doi:10.6342/NTU201701669

distributions, we minimize the cross entropy as

$$\begin{aligned} & \min \ H(\bar{\mathcal{L}}(z_{ik} \mid z_{jl}), q(z_{ik} \mid \bar{C}_t)) + H(\bar{\mathcal{L}}(z_{ik} \mid z_{jl}), q(z_{jl} \mid \bar{C}_{t'})) \\ = & \min \ -\hat{\mathcal{L}}(z_{ik} \mid z_{jl}) \log q(z_{ik} \mid \bar{C}_t) - (1 - \hat{\mathcal{L}}(z_{ik} \mid z_{jl})) \log(1 - q(z_{ik} \mid \bar{C}_t)) \\ & -\hat{\mathcal{L}}(z_{ik} \mid z_{jl}) \log q(z_{jl} \mid \bar{C}_{t'}) - (1 - \hat{\mathcal{L}}(z_{ik} \mid z_{jl})) \log(1 - q(z_{jl} \mid \bar{C}_{t'})). \end{aligned} \tag{3.12}$$

Here we denote the value-based method as MUSE-Q.

3.2.3 Sense Selection Strategy

Besides learning algorithm, sense selection is necessary to obtain training data for reinforcement learning. Unlike typical reinforcement learning scenario that the reward signal is a fixed function, MUSE exploits sense representation learning module, which also learns from data, as reward signal. On the other hand, sense selection also crucially affect the learning target of representation learning model. Hence, a naive sense selection algorithm would substantially undermine the performance of such modularized system.

Given a fitness estimation for each sense, exploiting the greedy sense is the most popular strategy for clustering algorithms [6, 24] and hard-EM algorithms [10, 20] in literature. However, greedy algorithm would suffer from inadequate exploration that the model may be restricted to local optimum [22]. Although sampling technique has been proposed for unsupervised sense representation method with probabilistic modeling [9], exploration for value-based methods (including clustering methods [6]) is not well studied.

In addition, there are two incentives to conduct exploration. First, in early training stage when the fitness is not well estimated, it is desirable to explore underestimated senses. Second, due to high ambiguity in natural language, sometimes multiple senses in a word would fit in the same context. The dilemma between exploring sub-optimal choices and exploiting the optimal choice is called exploration-exploitation trade-off in reinforcement learning [22].

We introduce exploration mechanisms for sense selection for both policy gradient and Q-learning. For policy gradient, we sample the policy distribution to approximate the expectation in (3.4). Because of the flexible formulation of Q-learning, the following

doi:10.6342/NTU201701669

Algorithm 1: Learning Algorithm

for $w_i = C_t \in C$ **do**
  sample $w_j = C_{t'}(0 < |t' - t| \leq m)$;
  $z_{ik} = \text{select}(C_t, w_i)$;
  $z_{jl} = \text{select}(C_{t'}, w_j)$;
  optimize U, V by (3.3) for the sense representation module;
  optimize P, Q by (3.4) or (3.12) for the sense selection module;

classic exploration mechanisms are applied to sense selection:

- *Greedy*: selects the sense with the largest Q-value (no exploration).
- ϵ-*Greedy*: selects a random sense with ϵ probability, and adopts the greedy strategy otherwise [16].
- *Boltzmann*: samples the sense based on the Boltzmann distribution modeled by Q-value. We directly use (3.1) as the Boltzmann distribution for simplicity.

Compared to Greedy selection, ϵ-Greedy enables exploration with ϵ probability. However, exploration of ϵ-Greedy is conducted by a uniform sampling. In contrast, Boltzmann sampling takes the relative ordering of Q-value into account, which may achieve more effective exploration than ϵ-Greedy.

We note that Q-learning with Boltzmann sampling yields the same sampling process as policy gradient but different optimization objective. To our best knowledge, we are among the first to explore several exploration strategies for unsupervised sense embedding learning.

In the following sections, MUSE-Q-Greedy denotes the model using corresponding sense selection strategy for Q-learning.

3.2.4 Summary

In summary, to jointly train sense selection and sense representation modules, we first select a pair of collocated senses, z_{ik} and z_{jl}, based on the sense selection module with a selecting strategy (e.g. greedy), and then optimize the sense representation module and the sense selection module using the above derivations. Algorithm 1 summarizes the training procedure for proposed MUSE model.

doi:10.6342/NTU201701669

3.3 Unstable Factor and Stabilization Methods

3.3.1 Unstable Factor

Although a systematic learning algorithm has been proposed in Algorithm 1, we notice some potential disadvantages of such formulation. To elaborate, for a pair of states $\bar{C}_t, \bar{C}_{t'}$ and corresponding actions z_{ik}, z_{jl} in our MDP formulation, two different possible reward signals $\mathcal{L}(z_{jl} \mid z_{ik})$ or $\mathcal{L}(z_{ik} \mid z_{jl})$ may be induced due to directional architecture in skip-gram ($\mathcal{L}(z_{jl} \mid z_{ik}) \neq \mathcal{L}(z_{ik} \mid z_{jl})$) [4]. That says, despite the same policy should be adopted for a word given its context, during learning the reward signal may differ according to whether it is the target word or the sampled collocation. Hence, the phenomenon would undermine the stability of model training.

In terms of the ideal scenario for estimated Q-values and policy, distinct rewards due to the distinction of input/output state in sense representation module implies that sense selection module should also model such distinction in Q-values, while the same sense selection policy should still be adopted under the same context despite the distinctions. That says, under the same context, despite distinct Q-values between states, the relative ordering of Q-values among possible candidates in each state should remain the same to deliver the same policy. To accomplish such mechanism, we propose two methods that address the problem by either modeling or circumventing such distinction. Note that similar dilemma also exists for policy gradient method that sense selection under the same context would suffer from unstable rewards.

3.3.2 Value Function Factorization

To model such distinction, we factorize the original definition of states $\bar{C}_t$ into two different states $\hat{C}_t$ and $\check{C}_t$, indicating the input and output direction in the following representation learning model, respectively. Hence, a naive solution may model two distinct Q-values $\hat{q}(\cdot)$ and $\check{q}(\cdot)$ for each factorized states by separate parameters $\hat{Q}$ and $\check{Q}$. However, despite distinct states, the same policy should be executed for the two states $\hat{C}_t$ and $\check{C}_t$. As a result, the relative ordering of Q-value for any two actions z_{ik} and $z_{ik'}$ should be the same for the same context $\bar{C}_t$ regardless of factorized states $\hat{C}_t$ and $\check{C}_t$. Hence, without

doi:10.6342/NTU201701669

proper restriction, the relative ordering may not hold for the two distinct Q-values with separate parameters $\hat{Q}$ and $\check{Q}$. Here we denote the problem as a value function ordering constraint problem for reinforcement learning.

To solve the value function ordering constraint problem, we factorized the Q-value to two distinct components. The first component is reward model aware that generates different value according to factorized states $\hat{C}_t$ and $\check{C}_t$, where the action value is not modeled. The second component is reward model agnostic that generate unique action value for the same context $\bar{C}_t$ regardless of factorized states to satisfy the value function ordering constraint. Technically, we maintain two sets of parameters S and A for each component. The reward model aware value function $q_S(\hat{C}_t)$ and $q_S(\check{C}_t)$, or state value, is generated as (3.13).

$$q_S(\hat{C}_t) = S_{i0}^T \sum_{j \in \bar{C}_t} P_j; \quad q_S(\check{C}_t) = S_{i1}^T \sum_{j \in \bar{C}_t} P_j. \tag{3.13}$$

On the other hand, the reward model agnostic value function $q_A(z_{ik} \mid \bar{C}_t)$, or *advantage* value [25], is generated as (3.14).

$$q_A(z_{ik} \mid \bar{C}_t) = A_{ik}^T \sum_{j \in \bar{C}_t} P_j. \tag{3.14}$$

Finally, the final Q-value can be represented as the addition between factorized value functions as (3.15)

$$q(z_{ik} \mid \hat{C}_t) = \sigma(q_S(\hat{C}_t) + q_A(z_{ik} \mid \bar{C}_t)); \quad q(z_{ik} \mid \check{C}_t) = \sigma(q_S(\check{C}_t) + q_A(z_{ik} \mid \bar{C}_t)). \tag{3.15}$$

With such factorization, we bind the same advantage value to different states to satisfy the value function ordering constraint. Here we denote the original formulation modeling Q-value as MUSE-Q and the factorized version as MUSE-SA that factorizes the Q-value to state value and advantage value. The proposed method is technically similar to the dueling network for deep reinforcement learning [25], which also factorizes Q-value as state value and advantage value to increase the stability of Deep Q-Networks [16] training. Finally, note that this method is not applicable to policy gradient methods that models

doi:10.6342/NTU201701669

policy distribution explicitly, as the same policy is inherently used regardless of input/output state in the representation module.

3.3.3 One-sided Optimization

On the other hand, to circumvent the unstable reward signal to the same context $\bar{C}_t$, here we propose a simple technique that conduct optimization on only one end of the directional representation learning model. That says, for a pair of actions z_{ik} and z_{jl} with reward signal $\mathcal{L}(z_{jl} \mid z_{ik})$, we only optimize the sense selection for the input sense z_{ik}. Technically, for value-based method, we replace (3.12) with (3.16) that conducts optimization on the input sense only.

$$\min H(\hat{\mathcal{L}}(z_{jl} \mid z_{ik}), q(z_{ik} \mid \bar{C}_t)) \tag{3.16}$$

On the other hand, for policy gradient method, although the same policy distribution should be formed regardless of input/output state in representation module, the oscillating rewards may still undermine the stability of model training. Hence, we also propose one-sided optimization method for policy gradient method that only optimize sense selection for the input sense z_{ik} as (3.17).

$$\mathbb{E}_{z_{ik} \sim \pi(\cdot \mid \bar{C}_t)}[\log \bar{\mathcal{L}}(z_{jl} \mid z_{ik})]. \tag{3.17}$$

Here we denote one-sided optimization method for MUSE as $\overrightarrow{\text{MUSE}}$.

Finally, we note three algorithmic difference between the proposed value function factorization method and one-sided optimization method. First, in terms of formulation, the value function factorization method is tailored for value-based methods, whereas one-sided optimization is simple, light-weight, and can be generalized to policy gradient method. Second, in terms of parameter usage, value function factorization includes additional parameters to model state values, while one-sided optimization employs the same parameter set. Third, in turns of computation, one-sided optimization enjoys efficiency by circumventing sense selection optimization for the output direction in the representation learning model, whereas value function factorization may benefit from extra optimization from both input and output directions.

doi:10.6342/NTU201701669

Chapter 4

Experiments

We evaluate our proposed MUSE models in both quantitative and qualitative experiments.

4.1 Experimental Setup

Our model is trained on the April 2010 Wikipedia dump [26], which contains approximately 1 billion tokens. For fair comparison, we adopt the same vocabulary set as [8] and [6]. For preprocessing, we convert all words to their lower cases, apply the Stanford tokenizer and the Stanford sentence tokenizer [27], and remove all sentences with less than 10 tokens. The number of senses per word in Q is set to 3 for fair comparison with prior work [6].

In the experiments, the context window size is set to 5 ($|\bar{C}_t| = 11$). For the negative sampling distribution in (3.3), we use $(1/|Z_i|)$ word-level unigram as sense-level unigram for efficiency to train skip-gram with $|Z_i|$ senses for word w_i. We exploit the $\mathrm{word2vec}$ implementation released by Tensorflow [28] for representation module training with subsampling technique [4] to accelerate the training process. The learning rate is set to 0.025. The embedding dimension is 300. We initialize Q and V as zeros, and P and U from uniform distribution $[-\sqrt{1/100}, \sqrt{1/100}]$ such that each embedding has unit length in expectation [29]. Our model uses 25 negative senses for negative sampling in (3.3). We use $\epsilon = 5\%$ for ϵ-Greedy sense selection strategy

In optimization, we conduct mini-batch training with 2048 batch size using the following procedure: 1) select senses in the batch; 2) optimize U, V using stochastic training

doi:10.6342/NTU201701669

within the batch for efficiency; 3) optimize P, Q using mini-batch training for robustness.

4.2 Experiment 1: Contextual Word Similarity

To evaluate the quality of the learned sense embeddings, we compute the similarity score between word pair given their respective local contexts and compare with the human-judged score using Stanford's Contextual Word Similarities (SCWS) dataset [8]. Specifically, given a list of word pairs with corresponding contexts, $\{(w_i, \bar{C}_t, w_j, \bar{C}_{t'})\}$, we calculate the Spearman's rank correlation ρ between human-judged similarity and model similarity estimations. Two major contextual similarity estimations are introduced by [7]: AvgSimC and MaxSimC. AvgSimC is a *soft* measurement that addresses the contextual information with a probability estimation:

$$\text{AvgSimC}(w_i, \bar{C}_t, w_j, \bar{C}_{t'}) = \sum_{k=1}^{|Z_i|} \sum_{l=1}^{|Z_j|} \pi(z_{ik}|\bar{C}_t)\pi(z_{jl}|\bar{C}_{t'})d(z_{ik}, z_{jl}), \tag{4.1}$$

where $d(z_{ik}, z_{jl})$ refers to the cosine similarity between $U_{z_{ik}}$ and $U_{z_{jl}}$. AvgSimC weights the similarity measurement of each sense pair z_{ik} and z_{jl} by their probability estimations. On the other hand, MaxSimC is a *hard* measurement that only considers the most probable senses:

$$\text{MaxSimC}(w_i, \bar{C}_t, w_j, \bar{C}_{t'}) = d(z_{ik}, z_{jl}), \tag{4.2}$$

$$z_{ik} = \arg\max_{z_{ik'}} \pi(z_{ik'}|\bar{C}_t), z_{jl} = \arg\max_{z_{jl'}} \pi(z_{jl'}|\bar{C}_{t'}).$$

4.2.1 Experiment Results for Different Formulations

In this section, we investigate the experimental impact of different formulations upon the proposed MUSE model. Specifically, we are interested in the following questions:

1. Whether does the proposed stabilized formulations, MUSE-SA and $\overrightarrow{\text{MUSE}}$, consistently improve the original formulation MUSE?
2. Whether does the proposed smoothed policy gradient method MUSE-Smooth-Policy consistently improve the MUSE-Policy?
3. Whether does the proposed exploration techniques, ϵ-Greedy and Boltzmann, for

doi:10.6342/NTU201701669

Method	MaxSimC	AvgSimC
1) Original Formulation		
MUSE-Policy	66.0	65.9
MUSE-Smooth-Policy	66.9	67.6
MUSE-Q-Greedy	66.3	67.5
MUSE-Q-ϵ-Greedy	67.4	68.3
MUSE-Q-Boltzmann	67.6	68.5
2) Value Function Factorization		
MUSE-SA-Greedy	65.8	67.2
MUSE-SA-ϵ-Greedy	67.5†	68.5†
MUSE-SA-Boltzmann	**67.9**†	**68.9**†
3) One-Sided Optimization		
$\overrightarrow{\text{MUSE}}$-Policy	66.1†	67.4†
$\overrightarrow{\text{MUSE}}$-Smooth-Policy	67.8†	68.8†
$\overrightarrow{\text{MUSE}}$-Q-Greedy	66.3	68.3†
$\overrightarrow{\text{MUSE}}$-Q-$\epsilon$-Greedy	67.4	68.6†
$\overrightarrow{\text{MUSE}}$-Q-Boltzmann	**67.9**†	68.7†

Table 4.1: Spearman's rank correlation ρ x100 on SCWS dataset for different formulations of the proposed MUSE model. † denotes superior performance to the original formulation. Bold numbers denote the most competitive methods in each criterion.

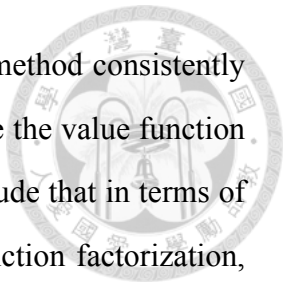
value-based methods consistently improve the Greedy method?

4. Whether does value-based method outperform policy gradient method in MUSE, or vice versa?

By answering the above questions, we examine the improvement brought by the proposed training techniques in terms of formulation, reward, and explorations. Besides, we also check the relative performance between value-based method and policy gradient. The experiment result is shown in Table 4.2, with † denoting improvement over the original formulation.

To answer the first question, we compare each entry in the original formulation with corresponding one in value function factorization and one-sided optimization in Table 4.1. We note that value function factorization does not support policy gradient method, so we do not compare it against policy gradient methods. Clearly, the value function factorization method improves the original formulation in 4 out of 6 settings, whereas the one-sided optimization method improves the original formulation in 8 out of 10 settings. On the other hand, in terms of the robustness of improvement, it is better to conduct non-inferiority trial

doi:10.6342/NTU201701669



instead of superiority trial. In this case, the one-sided optimization method consistently achieves non-inferiority to the original formulation in all cases, while the value function factorization remains non-inferiority in 4 out of 6 settings. We conclude that in terms of robust improvement, one-sided optimization is better than value function factorization, and both methods improves the original formulation in more than half settings.

To answer the second question about the improvement of MUSE-Smooth-Policy over MUSE-Policy, we compare the smoothed policy gradient method in each formulation (i.e., original formulation and one-sided optimization) with the corresponding vanilla policy gradient method. Significant improvement is observed consistently that Spearman's rank correlation is increased by at least 0.9 and at most 1.7 in all cases. This comparison demonstrates the effectiveness of reward smoothing adopted by MUSE-Smooth-Policy.

To answer the third question about the improvement of exploration techniques for value-based methods, we compare the exploration techniques, ϵ-Greedy and Boltzmann, in each formulation (i.e., original formulation, value function factorization, and one-sided optimization) with the corresponding Greedy strategy. Consistent improvement is observed that Spearman's rank correlation is increased by at least 0.3 and at most 2.1 in all cases. Besides, Boltzmann sampling also yields consistent improvement over ϵ-Greedy, validating our hypothesis in § 3.2.3 that Boltzmann sampling is more effective than ϵ-Greedy.

To answer the final question about the which of the proposed learning frameworks, value-based and policy gradient, is better, we compare the two methods. The best value-based method, MUSE-SA-Boltzmann, achieves comparable performance with the best policy gradient method, $\overrightarrow{\text{MUSE}}$-Smooth-Policy, in both evaluation measure. The phenomenon may be due to the same sampling strategy is adopted by both Boltzmann sampling and policy gradient methods. However, in the other cases, the value-based methods outperform the policy gradient methods in most settings, demonstrating the robustness of value-based methods that only optimizes the selected sense rather than optimizing all possible senses as policy gradient methods.

Besides, replacing $\bar{\mathcal{L}}(\cdot)$ with $\hat{\mathcal{L}}(\cdot)$ as the reward signal yields 2.3 times speedup for

doi:10.6342/NTU201701669

Method	MaxSimC	AvgSimC
Huang et al. (2012) [8]	26.1	65.7
Neelakantan et al. (2014)[6]	60.1	**69.3**
Tian et al. (2014)[19]	63.6	65.4
Li and Jurafsky (2015) [9]	66.6	66.8
Bartunov et al. (2016) [21]	53.8	61.2
Qiu et al. (2016) [10]	64.9	66.1
$\overrightarrow{\text{MUSE}}$-Policy	66.1	67.4
$\overrightarrow{\text{MUSE}}$-Smooth-Policy	67.8†	68.9
$\overrightarrow{\text{MUSE}}$-Q-Greedy	66.3	68.3
$\overrightarrow{\text{MUSE}}$-Q-$\epsilon$-Greedy	67.4†	68.6
$\overrightarrow{\text{MUSE}}$-Q-Boltzmann	**67.9**†	68.7

Table 4.2: Spearman's rank correlation ρ x100 on SCWS dataset for different models. † denotes superior performance to the state-of-the-art model in each criterion. Bold numbers denote the most competitive methods in each criterion.

MUSE-Q-ϵ-Greedy and 1.3 times speedup for $\overrightarrow{\text{MUSE}}$-Q-Boltzmann to reach 67.0 in MaxSimC, which demonstrates the efficacy of proposed approximation $\hat{\mathcal{L}}(\cdot)$ over original $\bar{\mathcal{L}}(\cdot)$ in terms of convergence.

In conclusion, we observe reliable improvement of various proposed training techniques in terms of formulations, rewards, and exploration techniques. Besides, despite the best performance of value-based method is comparable to that of policy gradient, value-based method is more reliable than policy gradient.

4.2.2 Experiment Results with the State-of-the-art

To validate the effectiveness of the proposed MUSE model, we also compare MUSE with the state-of-the-art baselines. For clarity purpose, we conduct experiments using one-sided optimization formulation of MUSE for comparison, since one-sided optimization is the most robust learning formulation according to the last section. We refer readers to Table 4.1 for detailed experiment results of MUSE.

The baselines for comparison include classic clustering methods [8, 6], EM algorithms [19, 10, 21], and Chinese Restaurant Process [9][1], where all approaches are trained on the same corpus except [10] used more recent Wikipedia dumps. The embedding sizes of all baselines are 300, except 50 in [8]. For every competitor with multiple settings, we report

[1] We run Li and Jurafsky's released code on our corpus for fair comparison [9].

doi:10.6342/NTU201701669

the best performance in each similarity measurement setting and show in Table 4.2.

Our MUSE model achieves the state-of-the-art performance on MaxSimC, demonstrating superior quality on independent sense embeddings. On the other hand, MUSE achieves comparable performance to the best competitor in terms of AvgSimC (68.9 vs. 69.3), while MUSE outperforms the same competitor significantly in terms of MaxSimC (67.9 vs. 60.1). The results demonstrate not only the high quality of sense representations but also accurate sense selection.

Finally, from the application perspective, MaxSimC refers to a typical scenario using single embedding per word, while AvgSimC employs multiple sense vectors simultaneously per word, which not only brings computational overhead but changes existing neural architecture for NLP. Hence, we argue that MaxSimC better characterize practical usage of a sense representation system than AvgSimC.

4.3 Experiment 2: Synonym Selection

We further evaluate our model on synonym selection using multi-sense word representations [20]. Evaluation is performed on three standard synonym selection datasets, ESL-50 [30], RD-300 [31], and TOEFL-80 [32]. In the datasets, each question consists of a question word w_Q and four answer candidates $\{w_A, w_B, w_C, w_D\}$, and the goal is to select the most semantically synonymous choice among the four candidates. For example, in the TOEFL-80 dataset, a question shows {(Q) enormously, (A) appropriately, (B) uniquely, (C) tremendously, (D) decidedly}, and the answer is (C). For multi-sense representations system, it selects the synonym of the question word w_Q using the maximum sense-level cosine similarity as a proxy of the semantic similarity [20].

For consistency, We following the last section to conduct experiments for the proposed $\overrightarrow{\text{MUSE}}$ model. Our model is compared with the following baselines:

1. Conventional word embeddings: global context vectors [8] and skip-gram [4];
2. Applying supervised word sense disambiguation using the IMS system and then applying skip-gram on disambiguated corpus (IMS+SG) [33];
3. Unsupervised sense embeddings: EM algorithm [20], multi-sense skip-gram (MSSG)

doi:10.6342/NTU201701669

Method	ESL-50 [30]	RD-300 [31]	TOEFL-80 [32]
1) Conventional Word Embedding			
Global Context [8]	47.73	45.07	60.87
Skip-Gram [4]	52.08	55.66	66.67
2) Word Sense Disambiguation			
IMS+SG [33]	41.67	53.77	66.67
3) Unsupervised Sense Embeddings			
EM [20]	27.08	33.96	40.00
MSSG [6]	57.14	58.93	78.26
CRP [9]	50.00	55.36	82.61
$\overrightarrow{\text{MUSE}}$-Policy	52.38	51.79	79.71
$\overrightarrow{\text{MUSE}}$-Smooth-Policy	57.14	**69.64**†	82.61
$\overrightarrow{\text{MUSE}}$-Q-Greedy	57.14	58.93	79.71
$\overrightarrow{\text{MUSE}}$-Q-$\epsilon$-Greedy	61.90†	62.50†	84.06†
$\overrightarrow{\text{MUSE}}$-Q-Boltzmann	**64.29**†	66.07†	**88.41**†
4) Supervised Sense Embeddings			
Retro-GC [20]	63.64	66.20	71.01
Retro-SG [20]	56.25	65.09	73.33

Table 4.3: Accuracy on synonym selection. † denotes superior performance to all unsupervised competitors

[6], Chinese Restaurant Process (CPR) [9], and the MUSE models;

4. Supervised sense embeddings with WordNet [34]: retrofitting global context vectors (Retro-GC) and retrofitting skip-gram (Retro-SG) [20].

Among unsupervised sense embedding approaches, CRP and MSSG refer to the baselines with highest MaxSimC and AvgSimC in Table 4.2 respectively. Here we report the setting for baselines based on the best average performance in this task. We also show the performance of supervised sense embeddings as an upperbound of unsupervised methods due to the usage of additional supervised information from WordNet.

The results are shown in Table 4.3, where † denotes superior performance to all unsupervised competitors. The MUSE methods with proper exploration ($\overrightarrow{\text{MUSE}}$-Q-$\epsilon$-Greedy and $\overrightarrow{\text{MUSE}}$-Q-Boltzmann) outperform all unsupervised baselines consistently, echoing the superior quality of our sense vectors in last section. In addition, $\overrightarrow{\text{MUSE}}$-Q-Boltzmann and $\overrightarrow{\text{MUSE}}$-Smooth-Policy also outperforms the supervised sense embeddings except 1 setting without any supervised signal during training.

doi:10.6342/NTU201701669

tie-1	context k-NN	braves finish the season in **tie** with the los angeles dodgers scoreless otl shootout 6-6 hingis 3-3 7-7 0-0 6-3 untied
tie-2	context k-NN	his later years proudly wore **tie** with the chinese characters for pants trousers shirt juventus blazer socks anfield jerseys
blackberry -1	context k-NN	of the mulberry or the **blackberry** and minos sent him to cranberries maple vaccinium apricot apple blackberries
blackberry -2	context k-NN	of the large number of **blackberry** users in the us federal smartphones sap microsoft ipv6 smartphone linux-based
head-1	context k-NN	shells and/or high explosive squash **head** hesh and/or anti-tank venter thorax neck spear millimeters fusiform beachy
head-2	context k-NN	head was shaven to prevent **head** lice serious threat back then shaved thatcher loki thorax mao luthor chest pressure
head-3	context k-NN	appoint john pope republican as **head** of the new army of multi-party appoints unicameral beria appointed

Table 4.4: Qualitative analysis. Different word senses are selected by MUSE according to different context. The respective k-NN (sorted by collocation likelihood) senses are shown to indicate respective semantic meanings.

4.4 Qualitative Analysis

We further conduct qualitative analysis to check the semantic meanings of different senses learned by MUSE with k-nearest neighbors (k-NN) using sense representations. In addition, we provide contexts in the training corpus where the sense will be selected to validate the sense selection module. Table 4.4 shows the results.

The learned sense embeddings of the words “tie”, “blackberry”, and “head” clearly correspond to correct senses under different contexts. Specifically, the first sense of “tie” corresponds to the sense of being even, and the second sense corresponds to the sense of cloth; the first sense of “blackberry” corresponds to the sense of fruit, and the second sense corresponds to the sense of smartphone brand; the first sense of “head” corresponds to the sense of the upper part of an object, the second sense corresponds to the sense of the upper part of human, and the last sense corresponds to the sense of leader.

Since we address an unsupervised setting that learns sense embeddings from unannotated corpus, the discovered senses highly depend on the training corpus. Hence, from our manual inspection, it is common for our model to discover only two senses in a word, like “tie” and “blackberry”. However, we maintain our effort in developing unsupervised sense embeddings learning methods in this work, where the number of discovered sense

doi:10.6342/NTU201701669

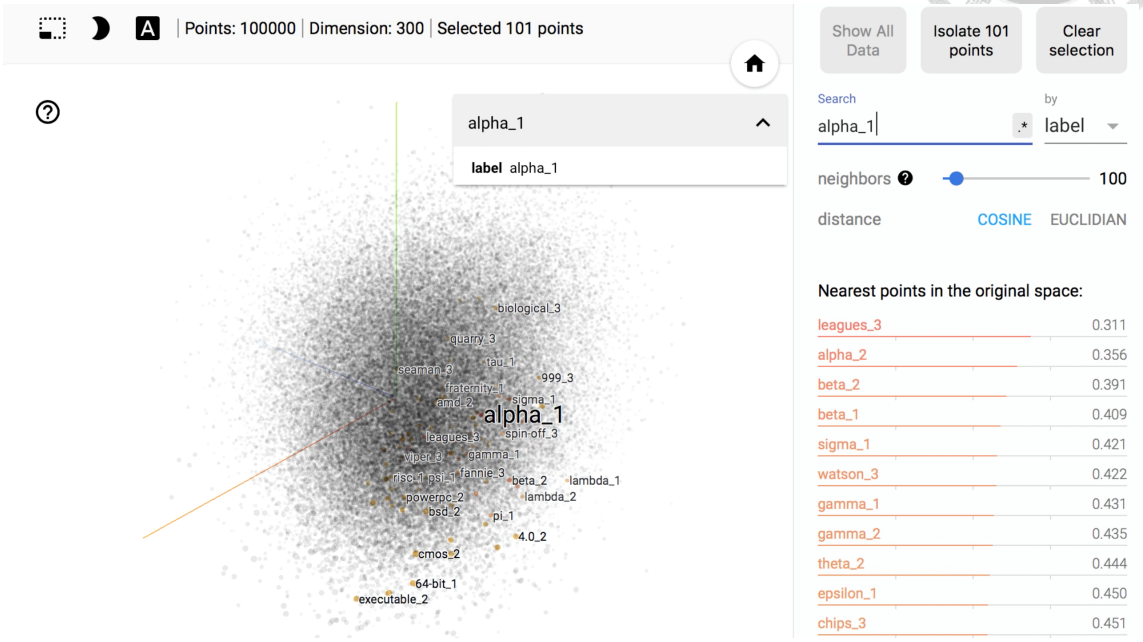
is not a focus. We refer readers to Manandhar et al.'s work for the task of unsupervised sense induction [35].

4.5 Visualization

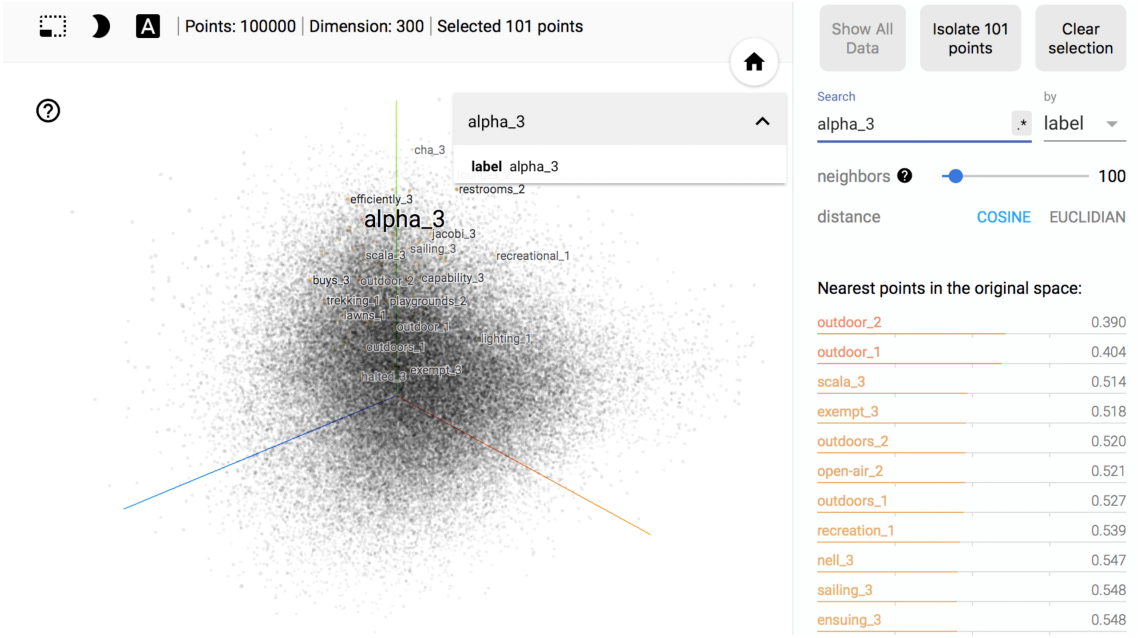
Finally, to visually demonstrate our system, we project our sense embeddings to 3-dimensional space using Principal Component Analysis (PCA) [1], and plot k-NN of different senses of words to illustrate the distinction between learned word senses.

Figure 4.1 and Figure 4.2 show the illustrations for sense-level k-NN of "alpha" and "directional", respectively. Specifically, the k-NN of one sense of "alpha" (Figure 4.1a) includes Greek letters like "gamma" and "lambda", while the k-NN of the other sense of "alpha" (Figure 4.1b) illustrates the semantics of brand names. On the other hand, one sense of "directional" (Figure 4.2a) refers to its physical aspect, which is revealed by its k-NN words like "frequency" and "signaling". In contrast, the other sense of "directional" (Figure 4.2b) depicts its mathematical aspect, as its k-NN words, such as "narrow" and "concave", describe mathematical properties. In sum, through visualization, we observe that minute distinction between senses, e.g., difference between physics and mathematics, can be discovered by MUSE.

doi:10.6342/NTU201701669



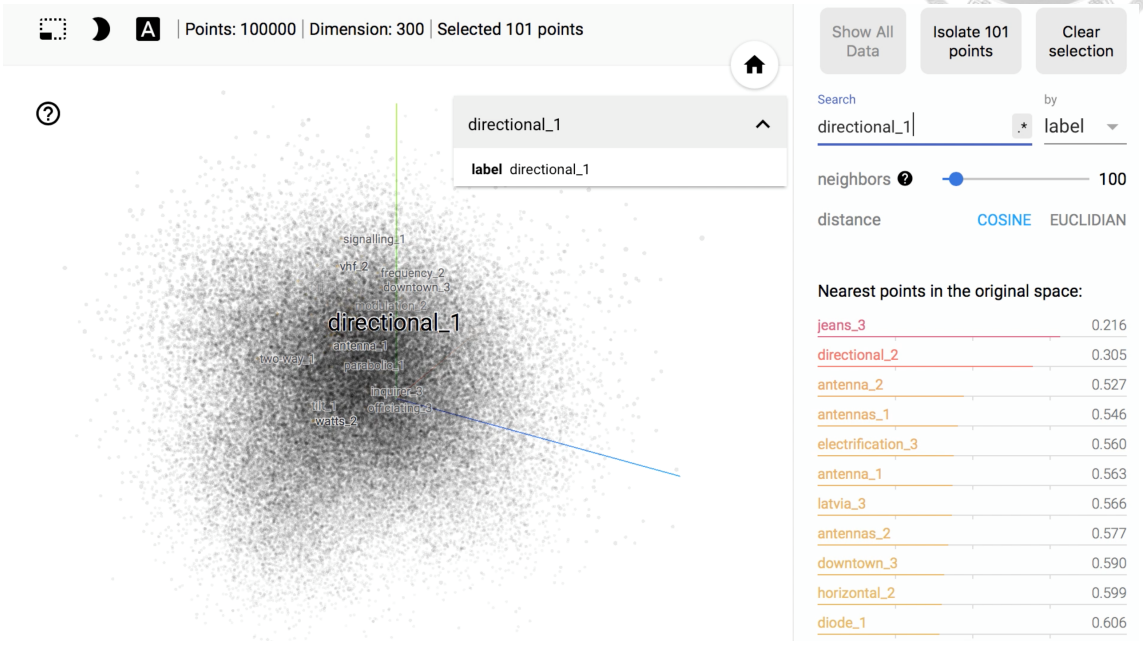
(a) K-NN of one sense of “alpha”



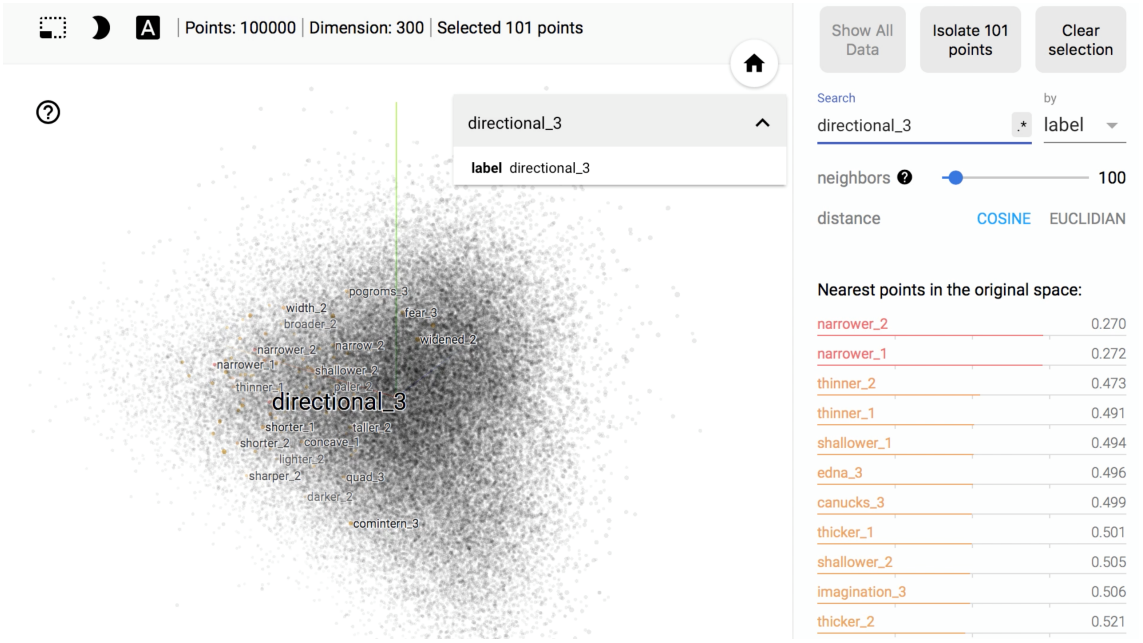
(b) K-NN of another sense of “alpha”

Figure 4.1: Visualization for k-NN of different senses of “alpha”. We exploit PCA [1] to project sense embeddings to 3-dimensional space.

doi:10.6342/NTU201701669



(a) K-NN of one sense of "directional"



(b) K-NN of another sense of "directional"

Figure 4.2: Visualization for k-NN of different senses of "directional". We exploit PCA [1] to project sense embeddings to 3-dimensional space.

doi:10.6342/NTU201701669

Chapter 5

Related Work

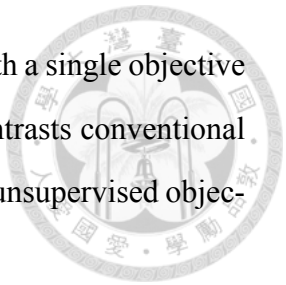
In this chapter, we review related work that is relevant to word sense representations. There are three dominant types of approaches for learning multi-sense word representations in the literature: 1) clustering methods, 2) probabilistic modeling methods, and 3) lexical ontology based methods. Our reinforcement learning based approach can be loosely connected to clustering methods and probabilistic modeling methods.

5.1 Clustering Methods

Reisinger and Mooney [7] first proposed multi-sense word representations on the vector space based on clustering techniques. With the power of deep learning, some work exploited neural networks to learn embeddings with sense selection based on clustering [8]. Neelakantan et al. [6] further exploits online clustering to learn sense representation in a non-parametric manner. Chen et al. [36] replaced the clustering procedure with a word sense disambiguation model using WordNet [34]. Kågebäck et al. [24] further exploited a weighting mechanism on contexts in the clustering procedure and evaluated their system on word sense induction. Vu and Parker [37] proposed an iterative process on the two-stage clustering-embedding learning framework. Moreover, Guo et al. [38] leveraged bilingual resources for clustering.

However, most of the above approaches separated the clustering procedure and the representation learning procedure without a joint objective, which may suffer from the error propagation issue. Instead, the proposed approach, MUSE, supports a joint learning algo-

doi:10.6342/NTU201701669



rithm that can optimize sense selection and representation modules with a single objective utilizing reinforcement learning. In sum, our modular framework contrasts conventional clustering methods as the first modularization framework with a joint unsupervised objective.

5.2 Probabilistic Methods

Instead of clustering, probabilistic modeling methods have been applied for learning multi-sense embeddings in order to make the sense selection more flexible, where Tian et al. [19] and Jauhar et al. [20] conducted probabilistic modeling with EM training. Li and Jurafsky [9] exploited Chinese Restaurant Process to infer the sense identity and demonstrates efficacy of multi-sense word representations on several downstream NLP tasks. Furthermore, Bartunov et al. [21] developed a non-parametric Bayesian extension on the skip-gram model [4] for multi-sense embeddings. Despite reasonable modeling on sense selection, all above methods mixed word-level tokens during sense representation learning to mitigate the complicated computation in their EM algorithms [21], which may adulterate sense representations by ambiguous word representations. In contrast, the proposed MUSE model conducts representation learning in sense-level purely.

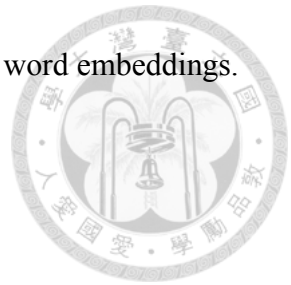
Recently, Qiu et al. [10] proposed an EM algorithm to learn purely sense-level representations, where the computational cost is high when decoding the sense identity sequence, because it takes exponential time to search all sense combination within a context window. Our modular design addresses such drawback, where the sense selection module decodes a sense sequence with linear-time complexity, while the sense representation module remains representation learning in the pure sense level.

5.3 Lexical Ontology Based Methods

Unlike a lot of relevant work that requires additional resources such as the lexical ontology [39, 40, 20, 41, 42] or bilingual data [38, 43, 44], which may be unavailable in some language, our model can be trained using only an unlabeled corpus. Also, some prior work proposed to learn topical embeddings and word embeddings jointly in order to consider the

doi:10.6342/NTU201701669

contexts [45, 46], whereas this paper focuses on learning multi-sense word embeddings.



doi:10.6342/NTU201701669

Chapter 6

Conclusion

This paper proposes a novel modularized framework for unsupervised sense representation learning, which supports not only flexible design on modular tasks but also joint optimization among modules. The proposed model is the first work that achieves purely sense-level representation learning with linear-time sense selection, and embodies hard decision of sense selection with a novel reinforcement learning algorithm. The experiments show that our *MUSE* model achieves the state-of-the-art performance on benchmark contextual word similarity and synonym selection tasks. Due to the unsupervised characteristics, the method is applicable to other language and even multi-lingual data. In the future, we would like to investigate semi-supervised word sense selection with the lexical ontology or word sense disambiguation data. We also plan to investigate reinforcement learning methods to incorporate multi-sense word representations for downstream NLP tasks.

doi:10.6342/NTU201701669

Bibliography

[1] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[3] T. K. Landauer, *Latent semantic analysis*. Wiley Online Library, 2006.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[5] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation.," vol. 14, pp. 1532–1543, 2014.

[6] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[7] J. Reisinger and R. J. Mooney, "Multi-prototype vector-space models of word meaning," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 109–117, Association for Computational Linguistics, 2010.

[8] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving Word Representations via Global Context and Multiple Word Prototypes," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

doi:10.6342/NTU201701669

[9] J. Li and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1722–1732, 2015.

[10] L. Qiu, K. Tu, and Y. Yu, "Context-dependent sense embedding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[11] G.-H. Lee and Y.-N. Chen, "MUSE: Modulizing unsupervised sense embeddings," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, 2013.

[13] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep neural network language models," in *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pp. 20–28, Association for Computational Linguistics, 2012.

[14] N.-Q. Pham, G. Kruszewski, and G. Boleda, "Convolutional neural network language models," in *Proc. of EMNLP*, 2016.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, vol. 2, p. 3, 2010.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *NIPS Deep Learning Workshop*, 2013.

[17] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," *Advances in neural information processing systems*, vol. 2, pp. 841–848, 2002.

doi:10.6342/NTU201701669

[18] F. Mosteller and J. W. Tukey, "Data analysis and regression: a second course in statistics.," *Addison-Wesley Series in Behavioral Science: Quantitative Methods*, 1977.

[19] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T.-Y. Liu, "A probabilistic model for learning multi-prototype word embeddings.," in *COLING*, pp. 151–160, 2014.

[20] S. K. Jauhar, C. Dyer, and E. H. Hovy, "Ontologically grounded multi-sense representation learning for semantic vector space models.," in *HLT-NAACL*, pp. 683–693, 2015.

[21] S. Bartunov, D. Kondrashkin, A. Osokin, and D. Vetrov, "Breaking sticks and ambiguities with adaptive skip-gram," *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, p. 130─138, 2016.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[23] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[24] M. Kågebäck, F. Johansson, R. Johansson, and D. Dubhashi, "Neural context embeddings for automatic discovery of word senses," in *Proceedings of NAACL-HLT*, pp. 25–32, 2015.

[25] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2016.

[26] C. Shaoul and C. Westbury, "The westbury lab wikipedia corpus," 2010.

[27] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60, 2014.

doi:10.6342/NTU201701669

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[29] T. Lei, R. Barzilay, and T. Jaakkola, "Molding cnns for text: non-linear, non-consecutive convolutions," 2015.

[30] P. D. Turney, "Mining the web for synonyms: Pmi-ir versus lsa on toefl," in *European Conference on Machine Learning*, pp. 491–502, Springer, 2001.

[31] M. Jarmasz and S. Szpakowicz, "Roget ' s thesaurus and semantic similarity," *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, vol. 2003, p. 111, 2004.

[32] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.," *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[33] Z. Zhong and H. T. Ng, "It makes sense: A wide-coverage word sense disambiguation system for free text," in *Proceedings of the ACL 2010 System Demonstrations*, pp. 78–83, Association for Computational Linguistics, 2010.

[34] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[35] S. Manandhar, I. P. Klapaftis, D. Dligach, and S. S. Pradhan, "Semeval-2010 task 14: Word sense induction & disambiguation," in *Proceedings of the 5th international workshop on semantic evaluation*, pp. 63–68, Association for Computational Linguistics, 2010.

[36] X. Chen, Z. Liu, and M. Sun, "A unified model for word sense representation and disambiguation.," in *EMNLP*, pp. 1025–1035, Citeseer, 2014.

[37] T. Vu and D. S. Parker, "K-embeddings: Learning conceptual embeddings for words using context," in *Proceedings of NAACL-HLT*, pp. 1262–1267, 2016.

doi:10.6342/NTU201701669

[38] J. Guo, W. Che, H. Wang, and T. Liu, "Learning sense-specific word embeddings by exploiting bilingual resources.," in *COLING*, pp. 497–507, 2014.

[39] M. T. Pilehvar and N. Collier, "De-conflated semantic representations," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[40] S. Rothe and H. Schütze, "Autoextend: Extending word embeddings to embeddings for synsets and lexemes," *arXiv preprint arXiv:1507.01127*, 2015.

[41] T. Chen, R. Xu, Y. He, and X. Wang, "Improving distributed representation of word sense via wordnet gloss composition and context clustering," Association for Computational Linguistics, 2015.

[42] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Sensembed: Learning sense embeddings for word and relational similarity.," in *ACL (1)*, pp. 95–105, 2015.

[43] A. Ettinger, P. Resnik, and M. Carpuat, "Retrofitting sense-specific word vectors using parallel text," in *Proceedings of NAACL-HLT*, pp. 1378–1383, 2016.

[44] S. Šuster, I. Titov, and G. van Noord, "Bilingual learning of multi-sense embeddings with discrete autoencoders," *NAACL-HLT 2016*, 2016.

[45] P. Liu, X. Qiu, and X. Huang, "Learning context-sensitive word embeddings with neural tensor skip-gram model.," in *IJCAI*, pp. 1284–1290, 2015.

[46] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings.," in *AAAI*, pp. 2418–2424, 2015.

doi:10.6342/NTU201701669