自然语言处理

5. 文本表示

虞剑飞

南京大学智能科学与技术学院

2025.3.19

本章内容

- 课程内容
  - 第一部分：NLP基础知识：概念、规则方法、文本分类与语言模型
    - 自然语言处理概述
      - 概要介绍自然语言处理的发展历史；
      - 典型应用简介；
      - 挑战问题；
      - 技术发展趋势。
    - 自然语言处理之规则方法
      - 规则方法的基本框架，以词形还原、分词等任务为例介绍规则方法的设计与实现过程。
    - 文本分类
      - 文本分类的应用；文本表示词袋模型；
      - 朴素贝叶斯文本分类模型；线性文本分类模型；
      - tf*idf表示模型；特征选择方法；
      - 文本分类评价方法。
    - 语言模型：概念、建模方法及应用
      - N-Gram语言模型；线性语言模型；模型参数估计；语言模型评价。

本章内容

- 课程内容
  - 第二部分：面向NLP的深度学习基础
    - 词向量
      - 词汇表征；潜在语义分析；词表示学习；
      - 词向量（word2vec）算法思想；
      - CBOW/Skip-Gram 模型学习和优化。
    - 文本卷积神经网络
      - 神经网络基础；
      - 卷积的种类与运算；
      - 面向文本的卷积神经网络；
      - 卷积神经网络的应用。
    - 循环神经网络
      - 循环神经网络语言模型；
      - GRU/LSTM等高级循环神经网络模型；递归神经网络；
      - 注意力机制；
      - 循环神经网络应用。
    - 高级神经网络与预训练模型
      - 自注意力机制；
      - Transformer网络结构；
      - 预训练模型框架简介。

本章内容

- 背景
- 向量空间模型的问题
- 表示学习模型

背景

- 文本
  - 文本是由文字和标点组成的字符串，短语、句子、段落和篇章都是不同粒度的文本
  - 字或字符组成词、短语，进而形成句子、段落和篇章

- 文本表示的必要性
  - 计算机进行文本理解，必须知道文本长什么样
  - 文本的形式化表示是反映文本内容和区分不同文本的有效途径

本章内容

- 背景
- 向量空间模型的问题
- 表示学习模型

向量空间模型

- 基本概念
  - 向量空间模型（vector space model, VSM）由G. Salton 等人于1960s末期在信息检索领域提出，核心是将文本视为特征项的集合
  - 特征项：VSM中最小的语言单元，可以是字、词、短语等。文本表示为特征项集合 $(t_1, t_2, \dots, t_n)$

Document	the	cat	sat	in	hat	with
the cat sat						
the cat sat in the hat						
the cat with the hat						

向量空间模型

- 基本概念
  - 向量空间模型（vector space model, VSM）由G. Salton 等人于1960s末期在信息检索领域提出，核心是将文本视为特征项的集合
  - 特征项权重：每个特征项在文本中的重要性不尽相同，用w表示特征项t的权重，相应地，文本可以表示为 $(t_1{:}\,w_1, t_2{:}\,w_2, \dots, t_n{:}\,w_n)$ 或 $(w_1, w_2, \dots, w_n)$

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

向量空间模型的问题

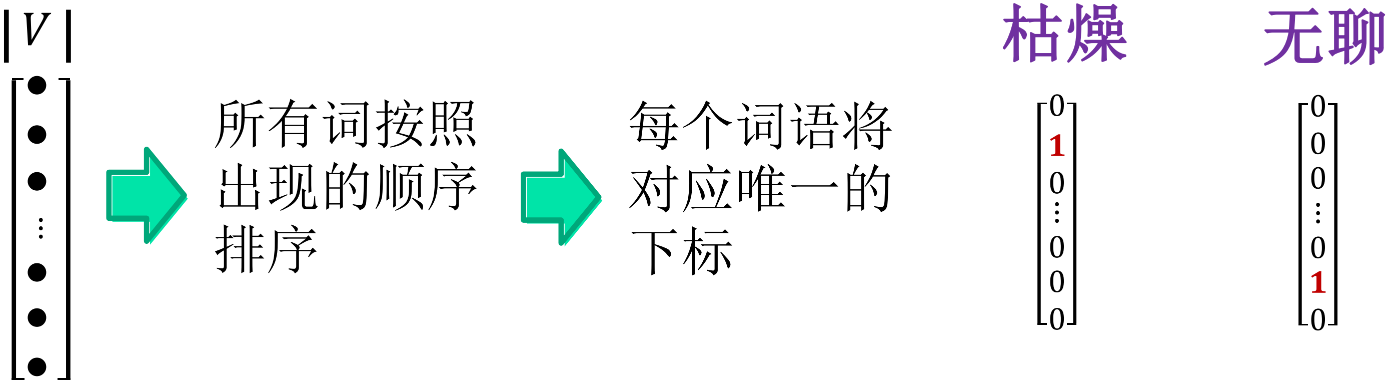
- 离散符号表示
  - 典型方法：抽象符号（字符串）

该 课程 很 **枯燥** ， 大家 觉得 很 **无聊** 。

w_0 =该　w_1=课程　w_2=很　w_3 =枯燥　w_4 =，

w_5=大家　w_6=觉得　w_7=很　w_8 =无聊　w_9 =。

  - 单词等价表示方法：one-hot表示法



$|V|$

所有词按照出现的顺序排序

每个词语将对应唯一的下标

枯燥

$$\begin{bmatrix}0\\1\\0\\\vdots\\0\\0\\0\end{bmatrix}$$

无聊

$$\begin{bmatrix}0\\0\\0\\\vdots\\0\\1\\0\end{bmatrix}$$

向量空间模型的问题

- 离散符号表示
  - 问题

$$\begin{array}{cc} \text{枯燥} & \text{无聊} \\ \begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{1} \\ 0 \end{bmatrix} \end{array}$$

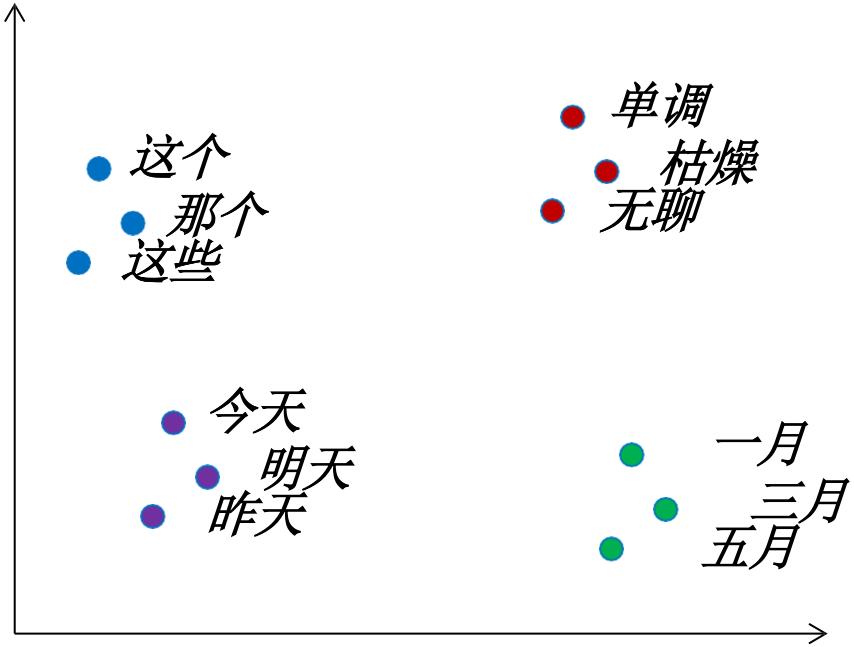
$$\text{枯燥} \otimes \text{无聊}$$

$$\begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{1} \\ 0 \end{bmatrix} = 0$$

⇨ 任意两个词之间的相似度都为**0**！

向量空间模型的问题

- 分布式表示



低维、稠密的连续实数空间

枯燥 $\begin{bmatrix} 0.24 \\ 0.15 \\ 0.42 \\ 0.51 \\ 0.21 \end{bmatrix}$

无聊 $\begin{bmatrix} 0.25 \\ 0.12 \\ 0.39 \\ 0.46 \\ 0.26 \end{bmatrix}$

本章内容

- 背景
- 向量空间模型的问题
- 表示学习模型

表示学习模型

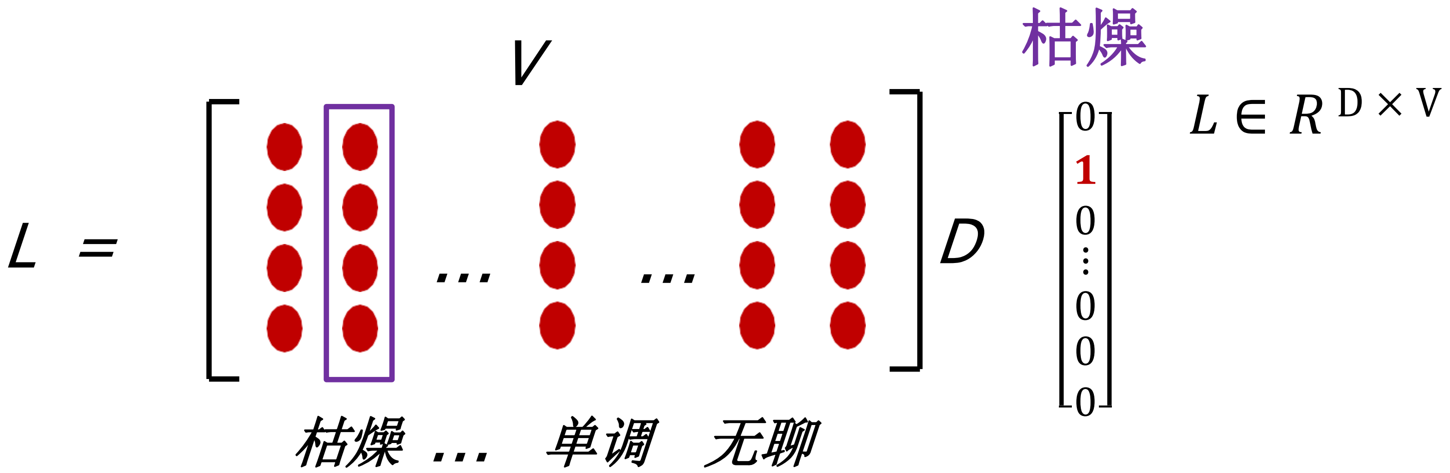
- **两种代表性学习方法**
  - 文本概念表示模型：以（概率）潜在语义分析（Latent Semantic Analysis, LSA）和潜在狄利克雷分布（Latent Dirichlet allocation, LDA）为代表的主题模型，旨在挖掘文本中的隐含主题或概念，文本将被表示为主题的分布向量
  - **深度表示学习模型**：通过深度学习模型以最优化特定目标函数（例如语言模型似然度）的方式在分布式向量空间中学习文本的低维实数向量表示

表示学习模型

- 表示学习模型
  - 词语的表示学习
  - 短语的表示学习
  - 句子的表示学习
  - 文档的表示学习

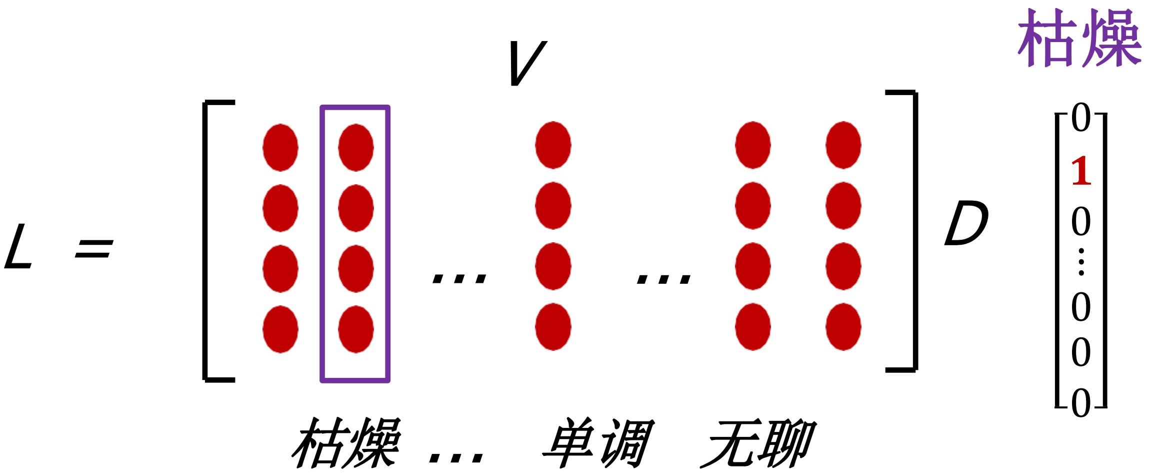
词语的表示学习

词向量表示



- 通常称为look-up table
  - 我们可以对L右乘一个词的one-hot表示 e得到该词的低维、稠密的实数向量表达：$x = Le$

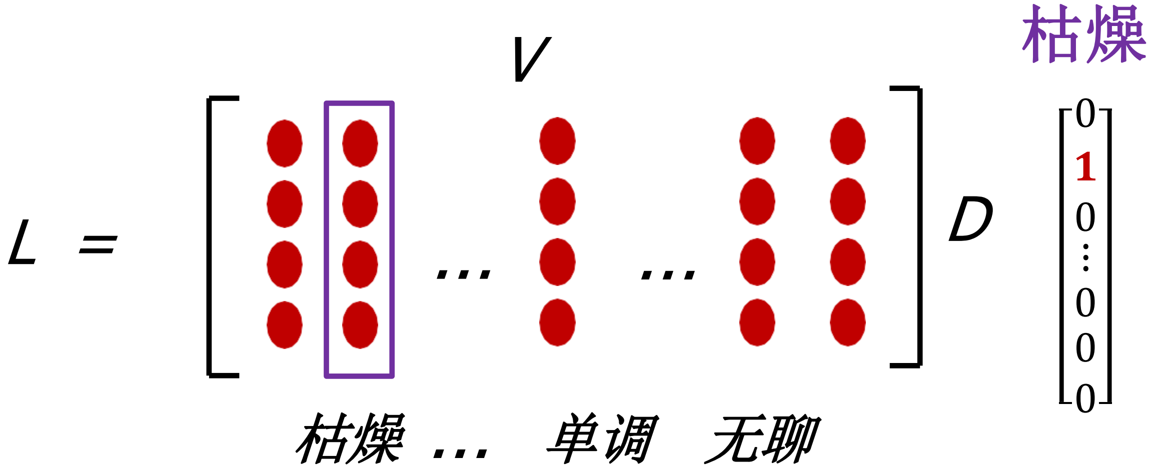
词语的表示学习



$L \in R^{\mathrm{D} \times \mathrm{V}}$

- 词表规模V和词向量维度D如何确定
  - V的确定：1. 训练数据中所有词；2. 频率高于某个阈值的所有词；3. 前V个频率最高的词
  - D的确定：超参数，人工设定，一般从几十到几百

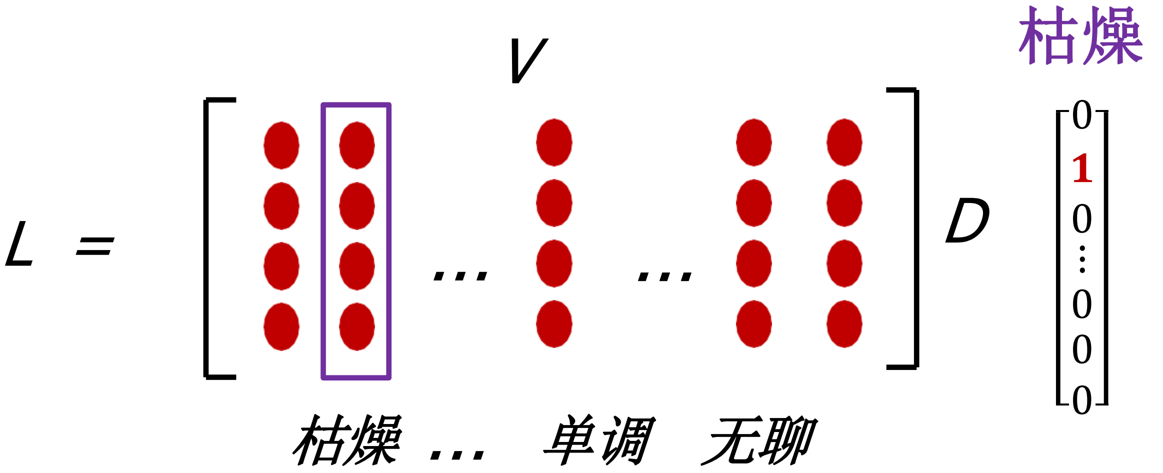
词语的表示学习



$L \in R^{\mathrm{D} \times \mathrm{V}}$

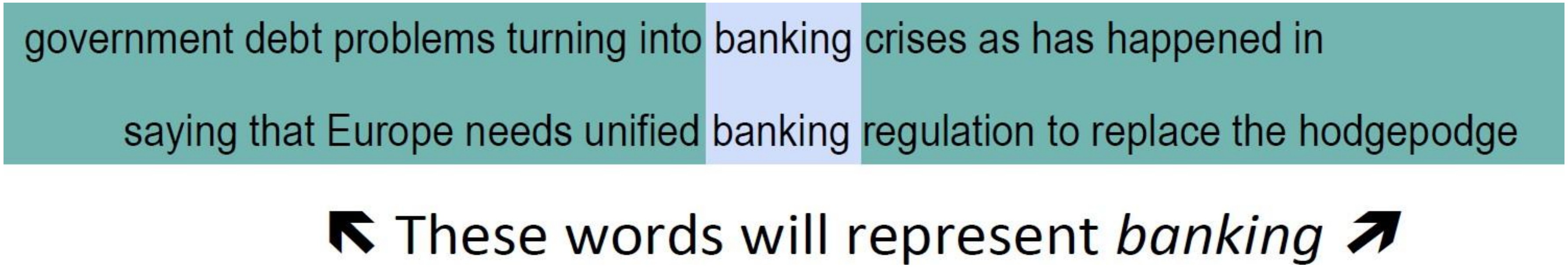
- 如何学习L
  - 通常先随机初始化，然后通过目标函数优化词的向量表达（e.g. 最大化语言模型似然度）

词语的表示学习



$L \in R^{D \times V}$

- 训练准则: ”You shall know a word by the company it keeps” (J. R. Firth 1957)



government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

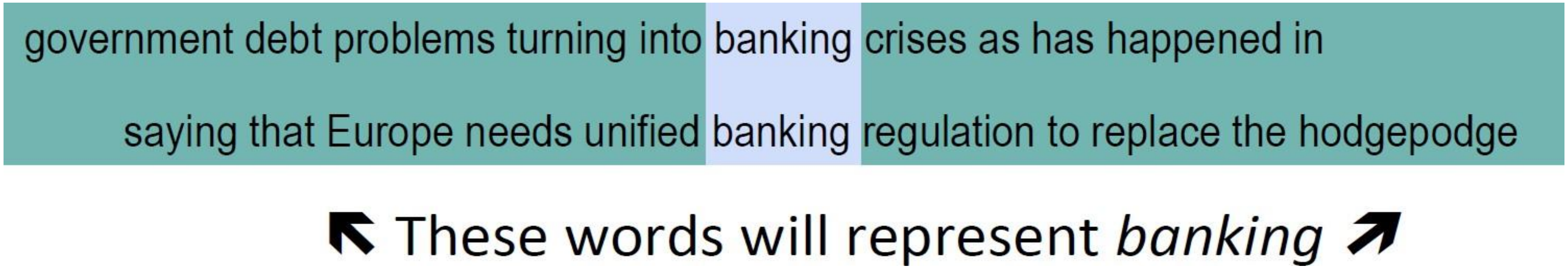
↖ These words will represent *banking* ↗

词语的表示学习-学习方法

- 表示学习模型
  - 词语的表示学习
    - 基于语言模型的方法
    - 直接学习法
      - C&W Model
      - CBOW and Skip-gram Model
      - GloVe
      - 字-词混合的表示学习

词语的表示学习-语言模型副产品

- 训练准则: ”You shall know a word by the company it keeps” (J. R. Firth 1957)



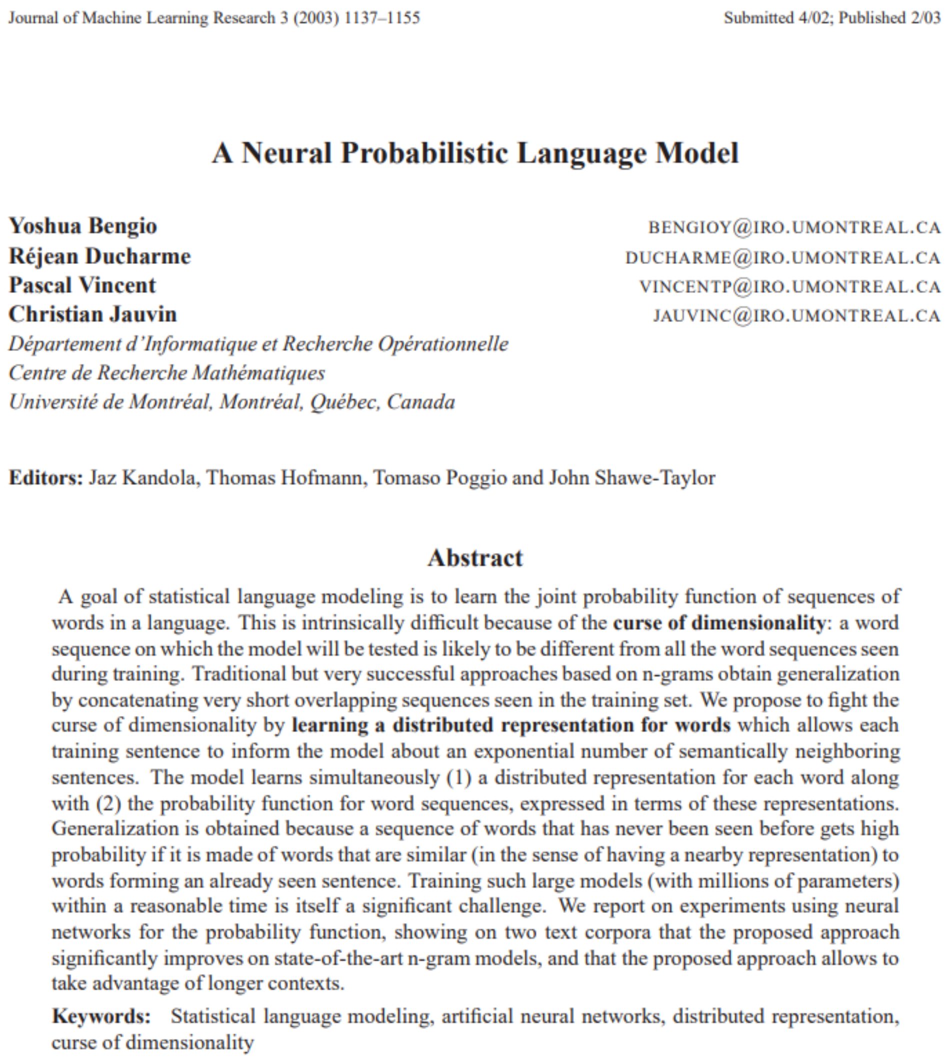
government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

$$P(w_1 w_2 \cdots w_{t-1} w_n) = \prod_{t=1}^{n} P(w_t | w_{t-1} \cdots w_{t-n+1})$$

词语的表示学习-语言模型副产品



Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03

A Neural Probabilistic Language Model

Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Réjean Ducharme DUCHARME@IRO.UMONTREAL.CA
Pascal Vincent VINCENTP@IRO.UMONTREAL.CA
Christian Jauvin JAUVINC@IRO.UMONTREAL.CA
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal, Montréal, Québec, Canada

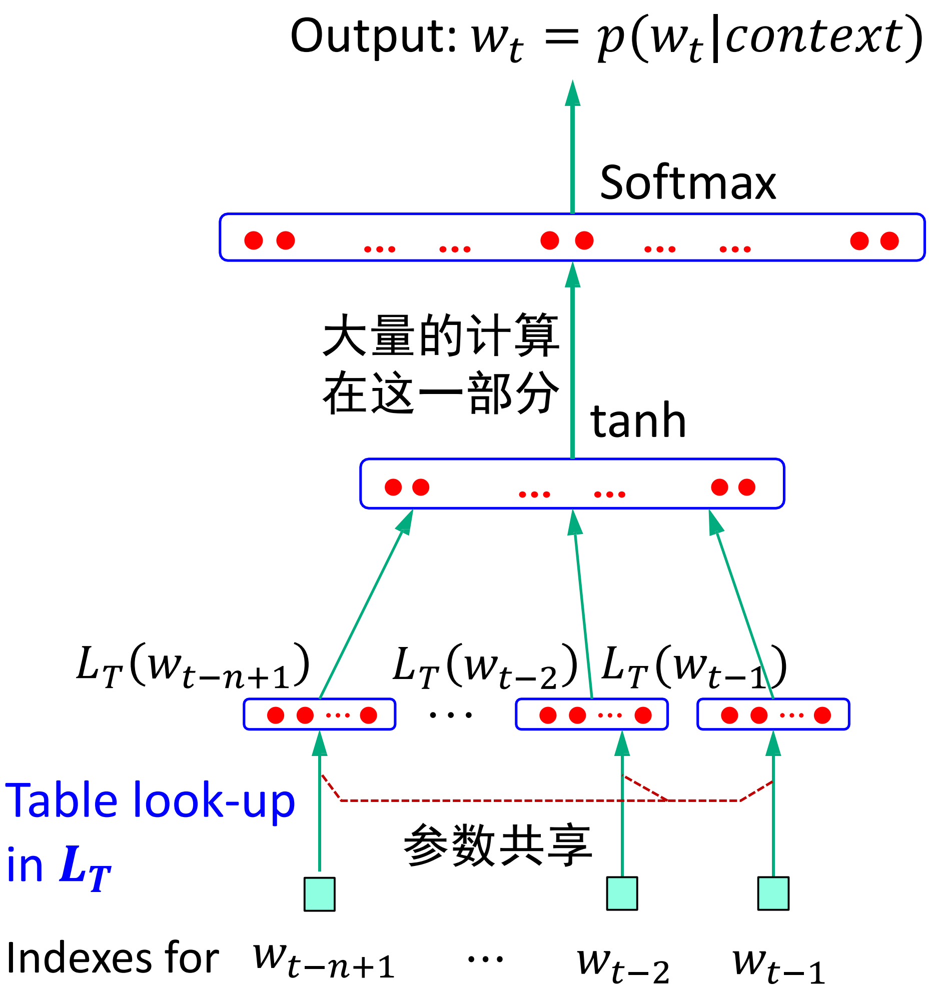
Editors: Jaz Kandola, Thomas Hofmann, Tomaso Poggio and John Shawe-Taylor

Abstract

A goal of statistical language modeling is to learn the joint probability function of sequences of words in a language. This is intrinsically difficult because of the **curse of dimensionality**: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training. Traditional but very successful approaches based on n-grams obtain generalization by concatenating very short overlapping sequences seen in the training set. We propose to fight the curse of dimensionality by **learning a distributed representation for words** which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. The model learns simultaneously (1) a distributed representation for each word along with (2) the probability function for word sequences, expressed in terms of these representations. Generalization is obtained because a sequence of words that has never been seen before gets high probability if it is made of words that are similar (in the sense of having a nearby representation) to words forming an already seen sentence. Training such large models (with millions of parameters) within a reasonable time is itself a significant challenge. We report on experiments using neural networks for the probability function, showing on two text corpora that the proposed approach significantly improves on state-of-the-art n-gram models, and that the proposed approach allows to take advantage of longer contexts.

Keywords: Statistical language modeling, artificial neural networks, distributed representation, curse of dimensionality

词语的表示学习-语言模型副产品



$$P(w_t|w_{t-1}\cdots w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

$$\theta \leftarrow \theta + \frac{\partial \log P}{\partial \theta}$$

$$y = U\tanh(Hx + d) + Wx + b$$

$$x = (L(w_{t-1}), \cdots L(w_{t-n+1}))$$

(Bengio et al., 2003)

词语的表示学习-学习方法

- 表示学习模型
  - 词语的表示学习
    - 基于语言模型的方法
    - 直接学习法
      - C&W Model
      - CBOW and Skip-gram Model
      - GloVe
      - 字-词混合的表示学习

词语的表示学习-C&W model

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning

Ronan Collobert COLLOBER@NEC-LABS.COM
Jason Weston JASONW@NEC-LABS.COM
NEC Labs America, 4 Independence Way, Princeton, NJ 08540 USA

Abstract

We describe a single convolutional neural network architecture that, given a sentence, outputs a host of language processing predictions: part-of-speech tags, chunks, named entity tags, semantic roles, semantically similar words and the likelihood that the sentence makes sense (grammatically and semantically) using a language model. The entire network is trained *jointly* on all these tasks using weight-sharing, an instance of *multitask learning*. All the tasks use labeled data except the language model which is learnt from unlabeled text and represents a novel form of *semi-supervised learning* for the shared tasks. We show how both *multitask learning* and *semi-supervised learning* improve the generalization of the shared tasks, resulting in state-of-the-art performance.

1. Introduction

The field of Natural Language Processing (NLP) aims to convert human language into a formal representation that is easy for computers to manipulate. Current end applications include information extraction, machine translation, summarization, search and human-computer interfaces.

While complete semantic understanding is still a far-distant goal, researchers have taken a divide and conquer approach and identified several sub-tasks useful for application development and analysis. These range from the syntactic, such as part-of-speech tagging, chunking and parsing, to the semantic, such as word-sense disambiguation, semantic-role labeling, named entity extraction and anaphora resolution. Currently, most research analyzes those tasks *separately*. Many systems possess few characteristics that would help develop a unified architecture which would presumably be necessary for deeper semantic tasks. In particular, many systems possess three failings in this regard: (i) they are *shallow* in the sense that the classifier is often linear, (ii) for good performance with a linear classifier they must incorporate many hand-engineered features specific for the task; and (iii) they cascade features learnt separately from other tasks, thus propagating errors.

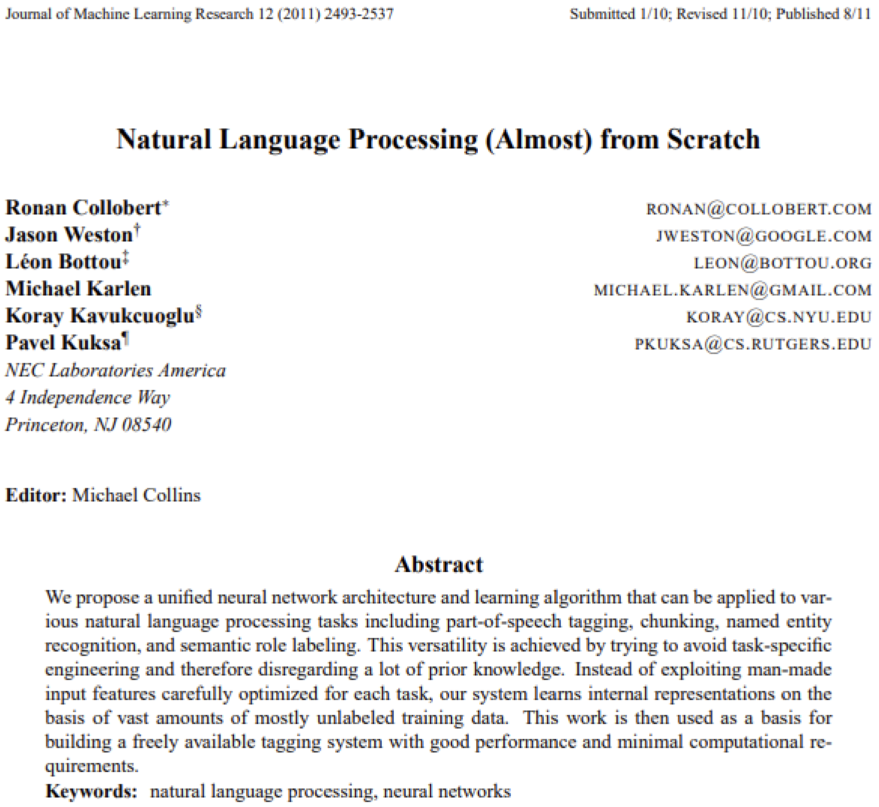
In this work we attempt to define a unified architecture for Natural Language Processing that *learns features* that are relevant to the tasks at hand given very limited prior knowledge. This is achieved by training a *deep neural network*, building upon work by (Bengio & Ducharme, 2001) and (Collobert & Weston, 2007). We define a rather general convolutional network architecture and describe its application to many well known NLP tasks including part-of-speech tagging, chunking, named-entity recognition, learning a language model and the task of semantic role-labeling.

All of these tasks are integrated into a single system which is trained *jointly*. All the tasks except the language model are supervised tasks with labeled training data. The language model is trained in an unsupervised fashion on the entire Wikipedia website. Training this task jointly with the other tasks comprises a novel form of *semi-supervised learning*.

We focus on, in our opinion, the most difficult of these tasks: the semantic role-labeling problem. We show that both (i) multitask learning and (ii) semi-supervised learning significantly improve performance on this task *in the absence of hand-engineered features*.

We also show how the combined tasks, and in particular the unsupervised task, learn powerful features with clear semantic information given no human supervision other than the (labeled) data from the tasks (see Table 1).

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).



Journal of Machine Learning Research 12 (2011) 2493-2537 Submitted 1/10; Revised 11/10; Published 8/11

Natural Language Processing (Almost) from Scratch

Ronan Collobert[*] RONAN@COLLOBERT.COM
Jason Weston[†] JWESTON@GOOGLE.COM
Léon Bottou[‡] LEON@BOTTOU.ORG
Michael Karlen MICHAEL.KARLEN@GMAIL.COM
Koray Kavukcuoglu[§] KORAY@CS.NYU.EDU
Pavel Kuksa[¶] PKUKSA@CS.RUTGERS.EDU
NEC Laboratories America
4 Independence Way
Princeton, NJ 08540

Editor: Michael Collins

Abstract

We propose a unified neural network architecture and learning algorithm that can be applied to various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. This versatility is achieved by trying to avoid task-specific engineering and therefore disregarding a lot of prior knowledge. Instead of exploiting man-made input features carefully optimized for each task, our system learns internal representations on the basis of vast amounts of mostly unlabeled training data. This work is then used as a basis for building a freely available tagging system with good performance and minimal computational requirements.

Keywords: natural language processing, neural networks

词语的表示学习-C&W model

$$(w_i, C) = w_{i-C}, \cdots, w_{i-1}, w_i, w_{i+1}, \cdots, w_{i+C}$$

we learned a **lot** from this lesson

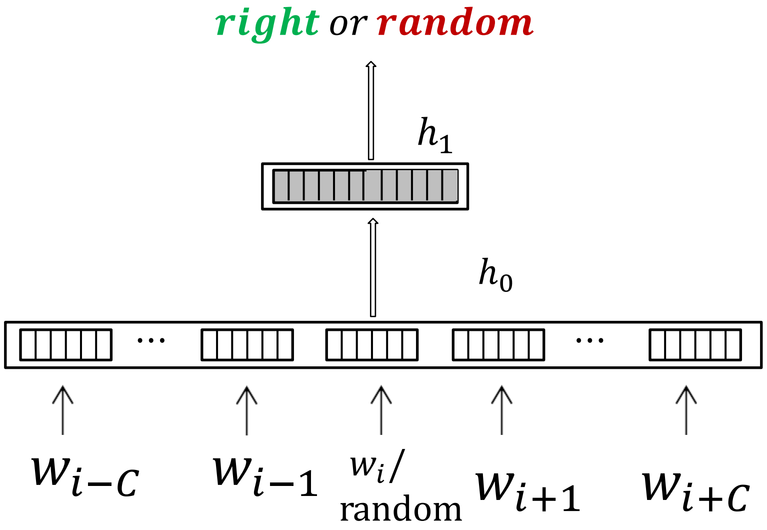
⇩ 随机替换

$$(w'_i, C) = w_{i-C}, \cdots, w_{i-1}, w'_i, w_{i+1}, \cdots, w_{i+C}$$

we learned a **today** from this lesson

$$score(w_i, C) > score(w'_i, C)$$

词语的表示学习-C&W model



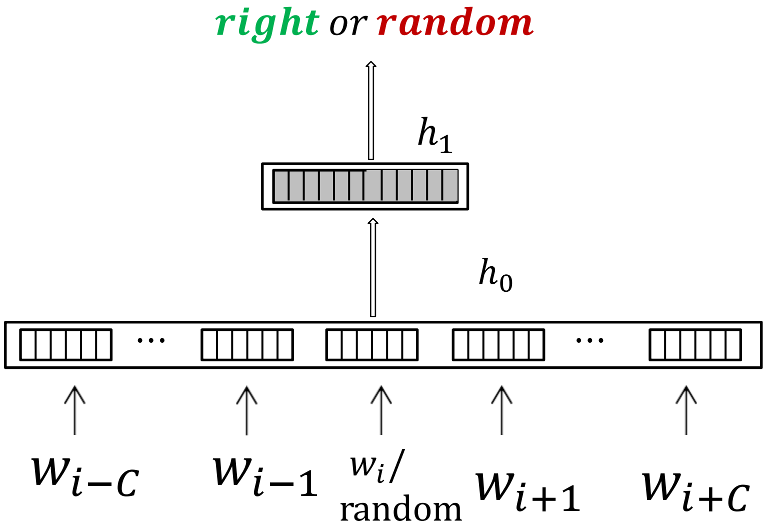
$$h_0 = [e(w_{i-C}), \cdots, e(w_{i-1}), e(w_i), e(w_{i+1}), \cdots, e(w_{i+C})]$$

$$h_1 = f(W_0 h_0 + b_0) \qquad score(w_i, C) = W_1 h_1 + b_1$$

$$h'_0 = [e(w_{i-C}), \cdots, e(w_{i-1}), e(w'_i), e(w_{i+1}), \cdots, e(w_{i+C})]$$

$$h'_1 = f(W_0 h'_0 + b_0) \qquad score(w'_i, C) = W_1 h'_1 + b_1$$

词语的表示学习-C&W model



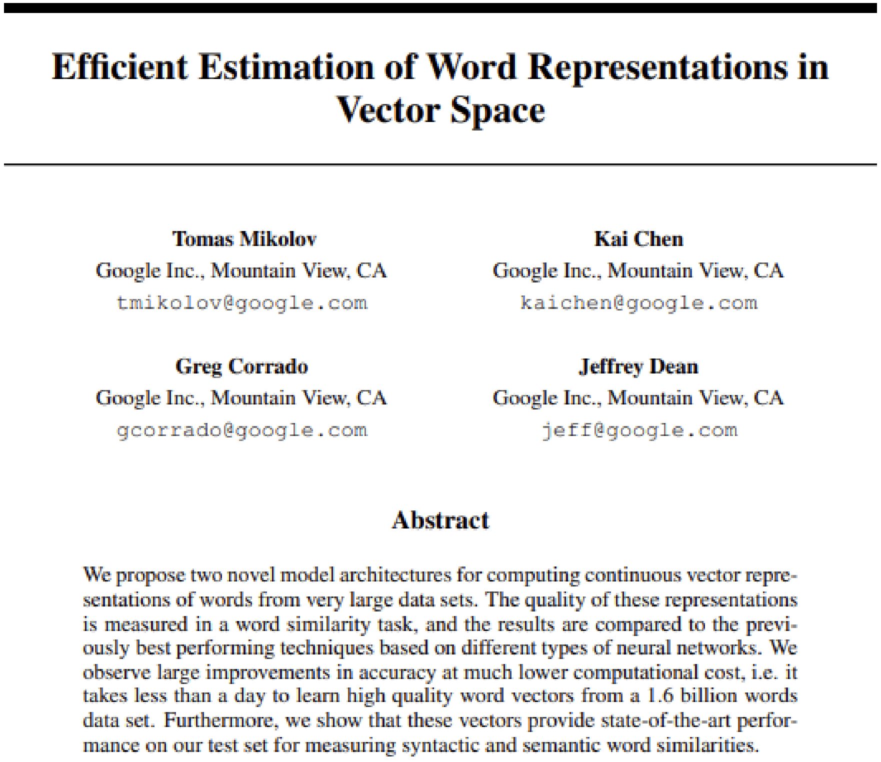
$$score(w_i, C) > score(w'_i, C) + 1$$

$$loss = \sum_{(w_i,C)\in D} \sum_{W'\in V'} max(0,1 + score(w'_i, C) - score(w_i, C))$$

词语的表示学习-学习方法

- 表示学习模型
  - 词语的表示学习
    - 基于语言模型的方法
    - 直接学习法
      - C&W Model
      - CBOW and Skip-gram Model
      - GloVe
      - 字-词混合的表示学习

词语的表示学习-CBOW model



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

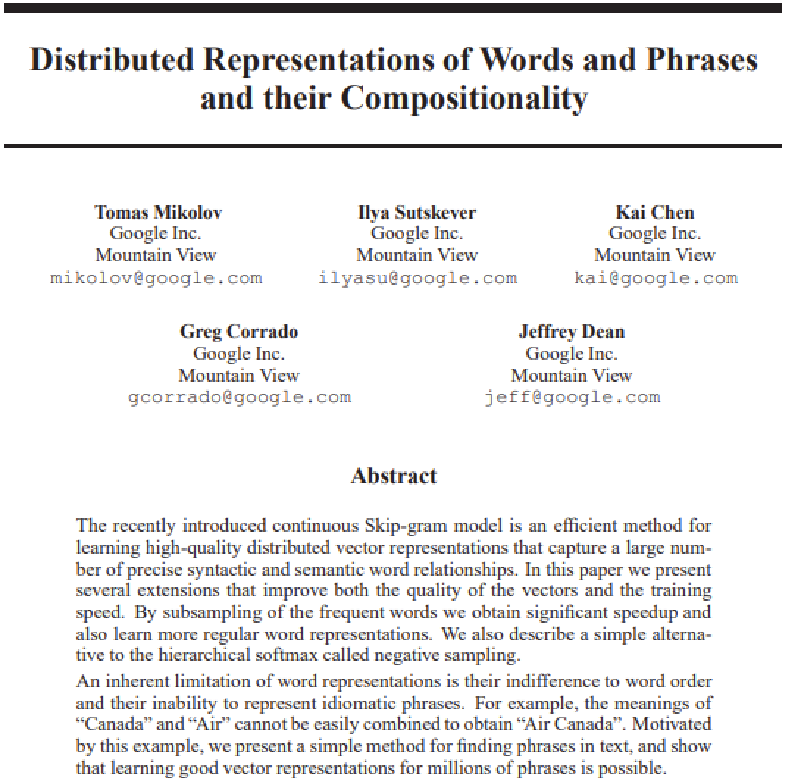
Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.



Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

Abstract

The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. In this paper we present several extensions that improve both the quality of the vectors and the training speed. By subsampling of the frequent words we obtain significant speedup and also learn more regular word representations. We also describe a simple alternative to the hierarchical softmax called negative sampling.

An inherent limitation of word representations is their indifference to word order and their inability to represent idiomatic phrases. For example, the meanings of "Canada" and "Air" cannot be easily combined to obtain "Air Canada". Motivated by this example, we present a simple method for finding phrases in text, and show that learning good vector representations for millions of phrases is possible.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, Jeffrey Dean:
Distributed Representations of Words and Phrases and their Compositionality. NIPS 2013: 3111-3119

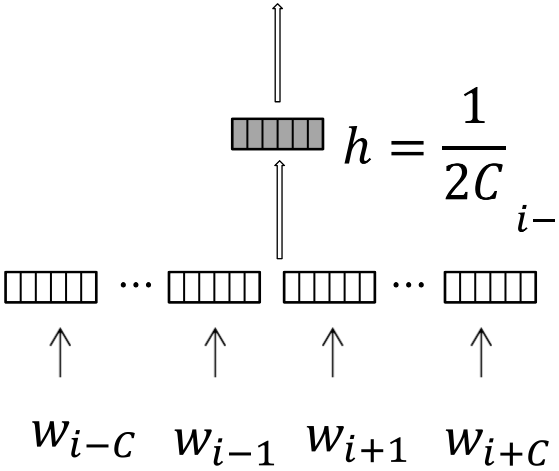
Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean:
Efficient Estimation of Word Representations in Vector Space. ICLR (Workshop Poster) 2013

词语的表示学习-CBOW model

- CBOW: Continuous Bag-of-Words
  - 词序不影响预测

$$w_{i-C},\cdots,w_{i-1},w_i,w_{i+1},\cdots,w_{i+C}$$

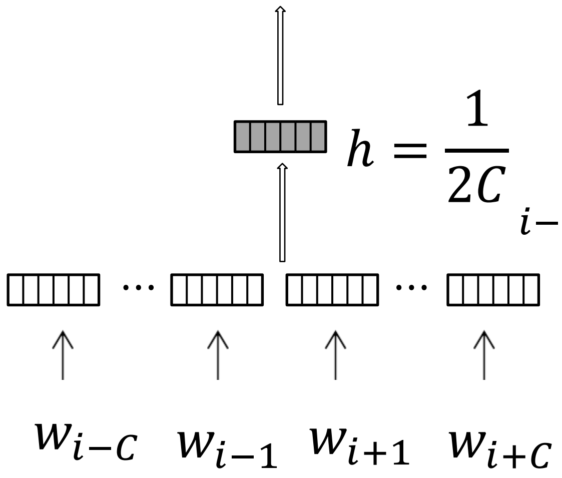
$$p(w_i|w_{i-C},\cdots,w_{i-1},w_{i+1},\cdots,w_{i+C})$$



$$h=\frac{1}{2C}\sum_{i-C\le k\le i+C,k\ne i}e(w_k)$$

词语的表示学习-CBOW model

$$p(w_i | w_{i-C}, \cdots, w_{i-1}, w_{i+1}, \cdots, w_{i+C})$$



$$h = \frac{1}{2C} \sum_{i-C \le k \le i+C, k \ne i} e(w_k)$$

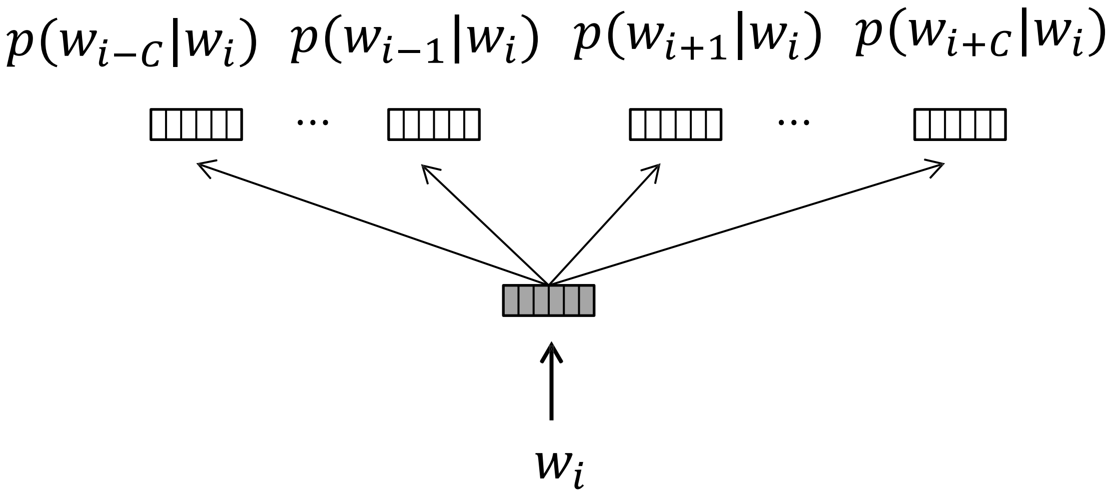
$$P(w_i | WC) = \frac{exp\{h \cdot e(w_i)\}}{\sum_{k=1}^{|V|} exp\{h \cdot e(w_k)\}}$$

$$L^* = \underset{L}{\operatorname{argmax}} \sum_{w_i} logP(w_i | WC)$$

词语的表示学习-Skip-gram model

- 不同于CBOW模型利用上下文词语预测中心词语，Skip-gram模型采用了相反的过程，即采用中心词语预测所有上下文词语。

$$w_{i-c}, \cdots, w_{i-1}, w_i, w_{i+1}, \cdots, w_{i+c}$$



$p(w_{i-c}|w_i)$ $p(w_{i-1}|w_i)$ $p(w_{i+1}|w_i)$ $p(w_{i+c}|w_i)$

$\cdots$ $\cdots$

w_i

词语的表示学习-Skip-gram model

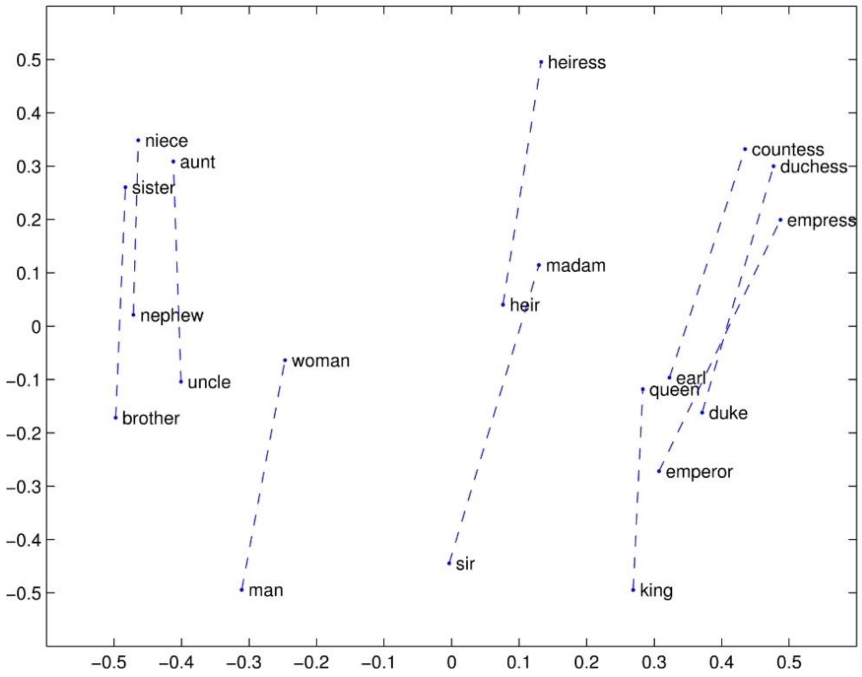
- 给定训练语料中任意一个n-元组$(w_i, C) = w_{i-C} \cdots w_{i-1} w_i w_{i+1} \cdots w_{i+C}$，Skip-gram模型直接利用中心词语$w_i$的词向量$e(w_i)$预测上下文$WC = w_{i-C} \cdots w_{i-1} w_{i+1} \cdots w_{i+C}$中每个词语$w_C$的概率:

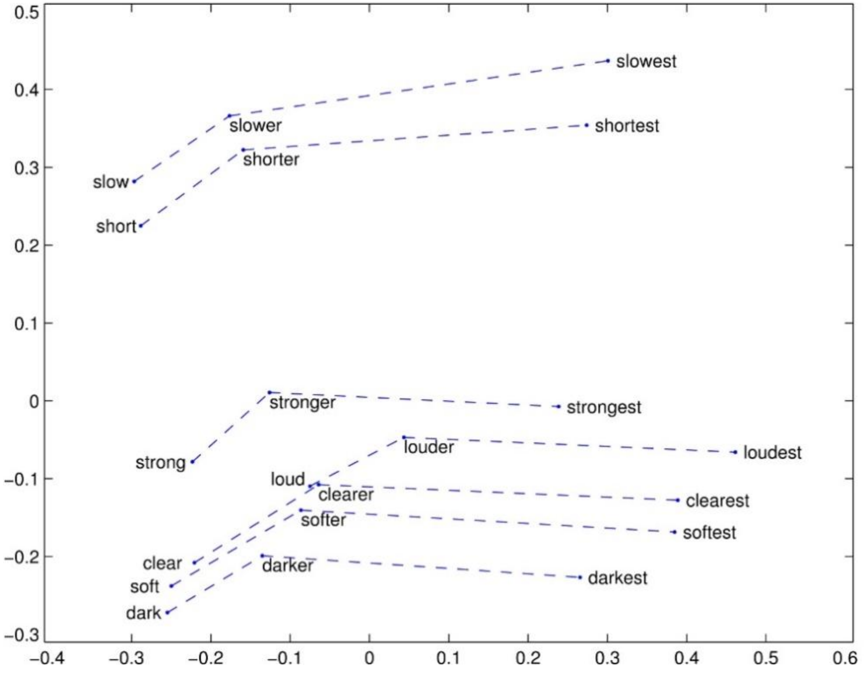
$$P(w_C|w_i) = \frac{exp\{e(w_i) \cdot e(w_C)\}}{\sum_{k=1}^{|V|} exp\{e(w_i) \cdot e(w_k)\}}$$

$$L^* = \operatorname*{argmax}_L \sum_{w_i \in V} \sum_{w_C \in WC} logP(w_C|w_i)$$

词语的表示学习-Skip-gram model

- 词向量可视化





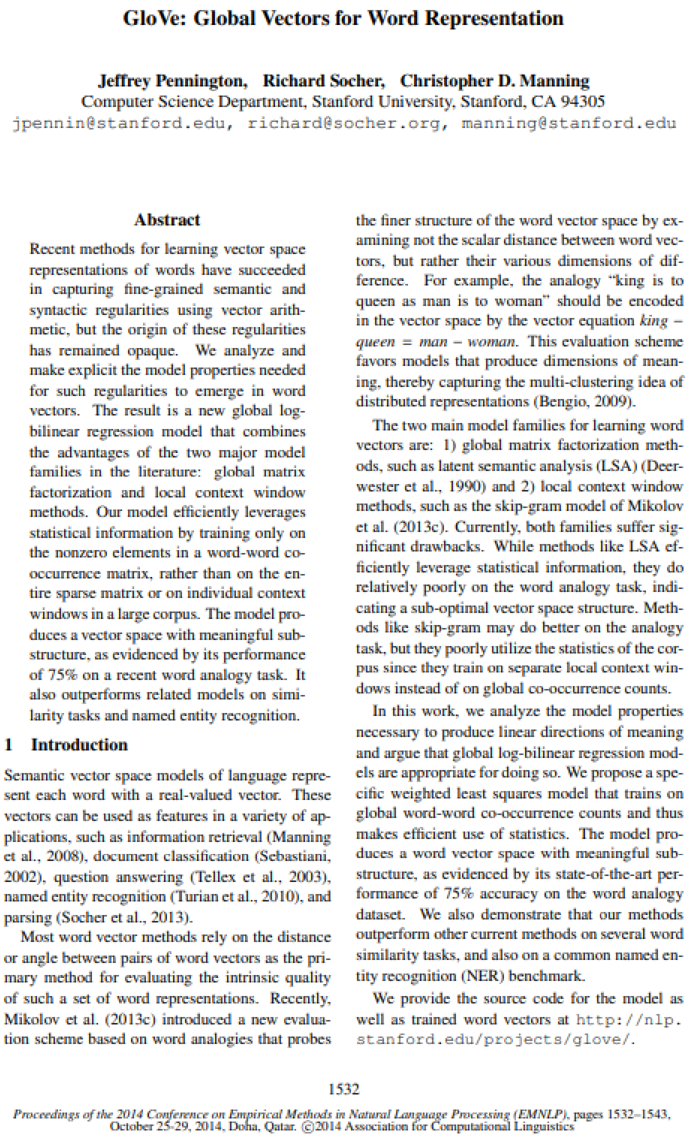
词语的表示学习-Skip-gram model

- 思考
  - CBOW和Skip-gram缺陷

词语的表示学习-学习方法

- 表示学习模型
  - 词语的表示学习
    - 基于语言模型的方法
    - 直接学习法
      - C&W Model
      - CBOW and Skip-gram Model
      - GloVe
      - 字-词混合的表示学习

词语的表示学习-GloVe



GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305
jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

Abstract

Recent methods for learning vector space representations of words have succeeded in capturing fine-grained semantic and syntactic regularities using vector arithmetic, but the origin of these regularities has remained opaque. We analyze and make explicit the model properties needed for such regularities to emerge in word vectors. The result is a new global log-bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization and local context window methods. Our model efficiently leverages statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. The model produces a vector space with meaningful substructure, as evidenced by its performance of 75% on a recent word analogy task. It also outperforms related models on similarity tasks and named entity recognition.

1 Introduction

Semantic vector space models of language represent each word with a real-valued vector. These vectors can be used as features in a variety of applications, such as information retrieval (Manning et al., 2008), document classification (Sebastiani, 2002), question answering (Tellex et al., 2003), named entity recognition (Turian et al., 2010), and parsing (Socher et al., 2013).

Most word vector methods rely on the distance or angle between pairs of word vectors as the primary method for evaluating the intrinsic quality of such a set of word representations. Recently, Mikolov et al. (2013c) introduced a new evaluation scheme based on word analogies that probes the finer structure of the word vector space by examining not the scalar distance between word vectors, but rather their various dimensions of difference. For example, the analogy "king is to queen as man is to woman" should be encoded in the vector space by the vector equation *king* − *queen* = *man* − *woman*. This evaluation scheme favors models that produce dimensions of meaning, thereby capturing the multi-clustering idea of distributed representations (Bengio, 2009).

The two main model families for learning word vectors are: 1) global matrix factorization methods, such as latent semantic analysis (LSA) (Deerwester et al., 1990) and 2) local context window methods, such as the skip-gram model of Mikolov et al. (2013c). Currently, both families suffer significant drawbacks. While methods like LSA efficiently leverage statistical information, they do relatively poorly on the word analogy task, indicating a sub-optimal vector space structure. Methods like skip-gram may do better on the analogy task, but they poorly utilize the statistics of the corpus since they train on separate local context windows instead of on global co-occurrence counts.

In this work, we analyze the model properties necessary to produce linear directions of meaning and argue that global log-bilinear regression models are appropriate for doing so. We propose a specific weighted least squares model that trains on global word-word co-occurrence counts and thus makes efficient use of statistics. The model produces a word vector space with meaningful substructure, as evidenced by its state-of-the-art performance of 75% accuracy on the word analogy dataset. We also demonstrate that our methods outperform other current methods on several word similarity tasks, and also on a common named entity recognition (NER) benchmark.

We provide the source code for the model as well as trained word vectors at http://nlp.stanford.edu/projects/glove/.

1532

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

词语的表示学习-GloVe

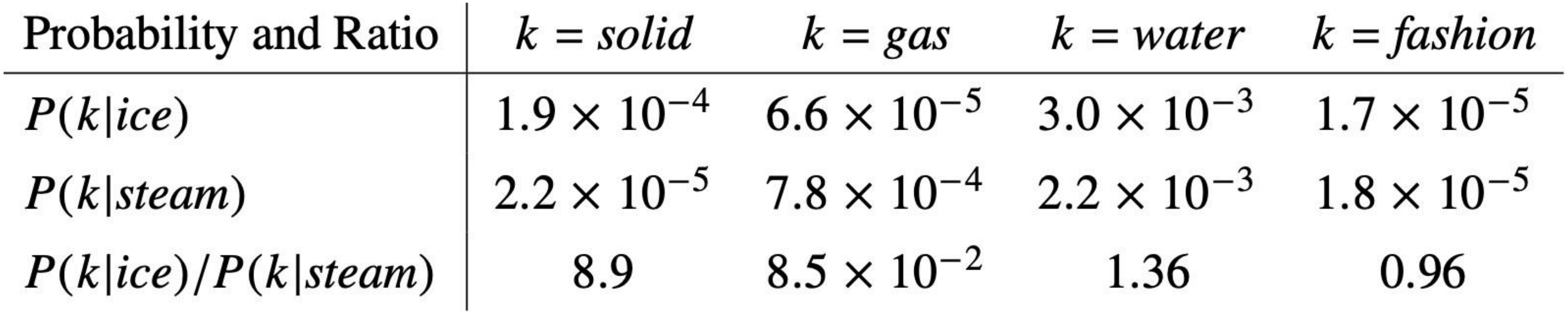
- C&W模型、CBOW以及Skip-gram模型都是采用局部上下文信息，没有用到语料的整体分布信息
- GloVe (Global Vectors for Word Representation) 模型旨在同时充分利用局部上下文信息和语料的整体分布信息，其主要思想如下：
  - 根据语料库构建一个共现矩阵，矩阵中的每一个元素X_{ij}代表单词j在单词i特定大小的上下文窗口内共同出现的次数
  - 构建词向量，目标使得词向量之间的关系能够反映共现矩阵之间的关系

词语的表示学习-GloVe

X_{ij}表示单词j出现在单词 i的上下文的次数；

X_i表示单词i的上下文中所有单词出现的总次数，即$X_i = \sum_k X_{ik}$；

$P_{ij} = P(i|j) = X_{ij}/X_i$表示单词$j$出现在单词 i的上下文中的概率；



Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

词语的表示学习-GloVe

$e(w_i)$，$e(w_j)$表示单词i和单词j的词向量表示；

$e(\widetilde{w}_k)$表示上下文单词的词向量表示；

$$F\left(e(w_i), e(w_j), e(\widetilde{w}_k)\right) = \frac{P_{ik}}{P_{jk}}$$

$$F\left(e(w_i) - e(w_j), e(\widetilde{w}_k)\right) = \frac{P_{ik}}{P_{jk}}$$

词语的表示学习-GloVe

$e(w_i)$，$e(w_j)$表示单词i和单词j的词向量表示；

$e(\widetilde{w}_k)$表示上下文单词的词向量表示；

$$F\big(e(w_i) - e(w_j), e(\widetilde{w}_k)\big) = \frac{P_{ik}}{P_{jk}}$$

$$F\left(\big(e(w_i) - e(w_j)\big)^T e(\widetilde{w}_k)\right) = \frac{P_{ik}}{P_{jk}}$$

词语的表示学习-GloVe

要求函数F在$(\mathbb{R},+)$和$(\mathbb{R}_{>0},\times)$两个群之间满足同态性

$$F\left(\left(e(w_i)-e(w_j)\right)^T e(\widetilde{w}_k)\right)=\frac{P_{ik}}{P_{jk}}$$

$$F\left(\left(e(w_i)-e(w_j)\right)^T e(\widetilde{w}_k)\right)$$

$$=F\left(e(w_i)^T e(\widetilde{w}_k)-e(w_j)^T e(\widetilde{w}_k)\right)=\frac{F\left(e(w_i)^T e(\widetilde{w}_k)\right)}{F\left(e(w_j)^T e(\widetilde{w}_k)\right)}$$

⇩

$$F=exp$$

$$F(e(w_i)^T e(\widetilde{w}_k))=P_{ik}=\frac{X_{ik}}{X_i}$$

词语的表示学习-GloVe

要求函数F在$(\mathbb{R},+)$和$(\mathbb{R}_{>0},\times)$两个群之间满足同态性

内积$e(w_i)^T e(\widetilde{w}_k)$满足对称性，将$\log(X_i)$设置为一个偏置项

$$F\left(e(w_i)^T e(\widetilde{w}_k)\right) = P_{ik} = \frac{X_{ik}}{X_i}$$

$$e(w_i)^T e(\widetilde{w}_k) = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$e(w_i)^T e(\widetilde{w}_k) + b_i + \tilde{b}_k = \log(X_{ik})$$

词语的表示学习-GloVe

要求函数F在$(\mathbb{R},+)$和$(\mathbb{R}_{>0},\times)$两个群之间满足同态性

内积$e(w_i)^T e(\widetilde{w}_k)$满足对称性，将$\log(X_i)$设置为一个偏置项

不同共现频率应拥有不同的权重

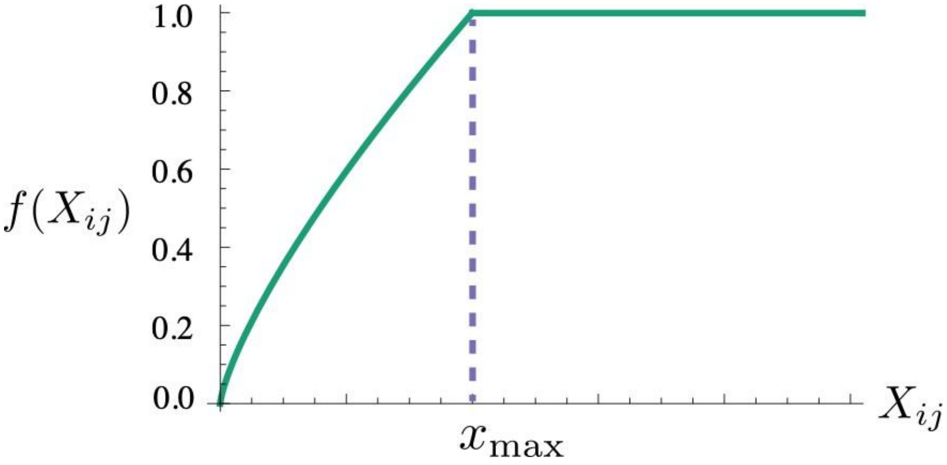
$$e(w_i)^T e(\widetilde{w}_k) + b_i + \tilde{b}_k = \log(X_{ik})$$

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(e(w_i)^T e(\widetilde{w}_j) + b_i + \tilde{b}_j - \log(X_{ij})\right)^2$$

词语的表示学习-GloVe

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(e(w_i)^T e(\widetilde{w}_j) + b_i + \tilde{b}_j - \log(X_{ij})\right)^2$$

$$f(X_{ij}) = \begin{cases} (x/x_{max})^{\alpha} & if\ x < x_{max} \\ 1 & otherwise \end{cases}$$



$x_{max} = 100$

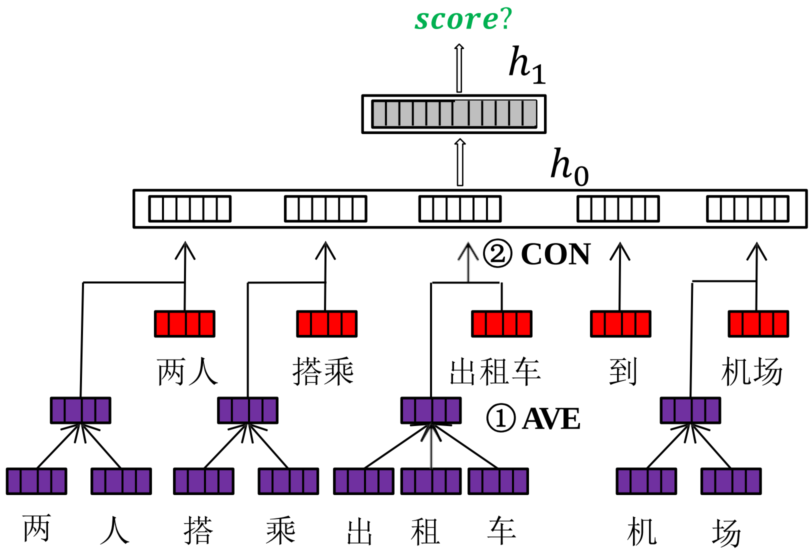
$\alpha = 3/4$

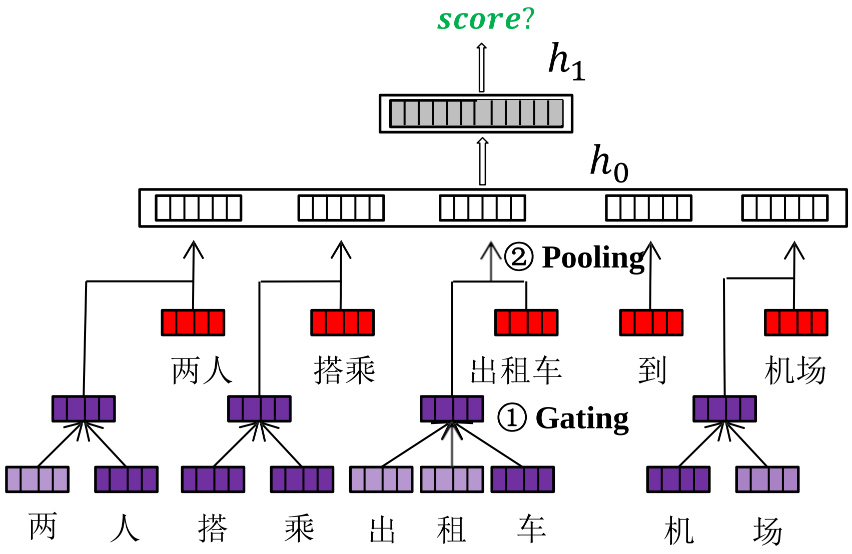
词语的表示学习-学习方法

- 表示学习模型
  - 词语的表示学习
    - 基于语言模型的方法
    - 直接学习法
      - C&W Model
      - CBOW and Skip-gram Model
      - GloVe
      - 字-词混合的表示学习

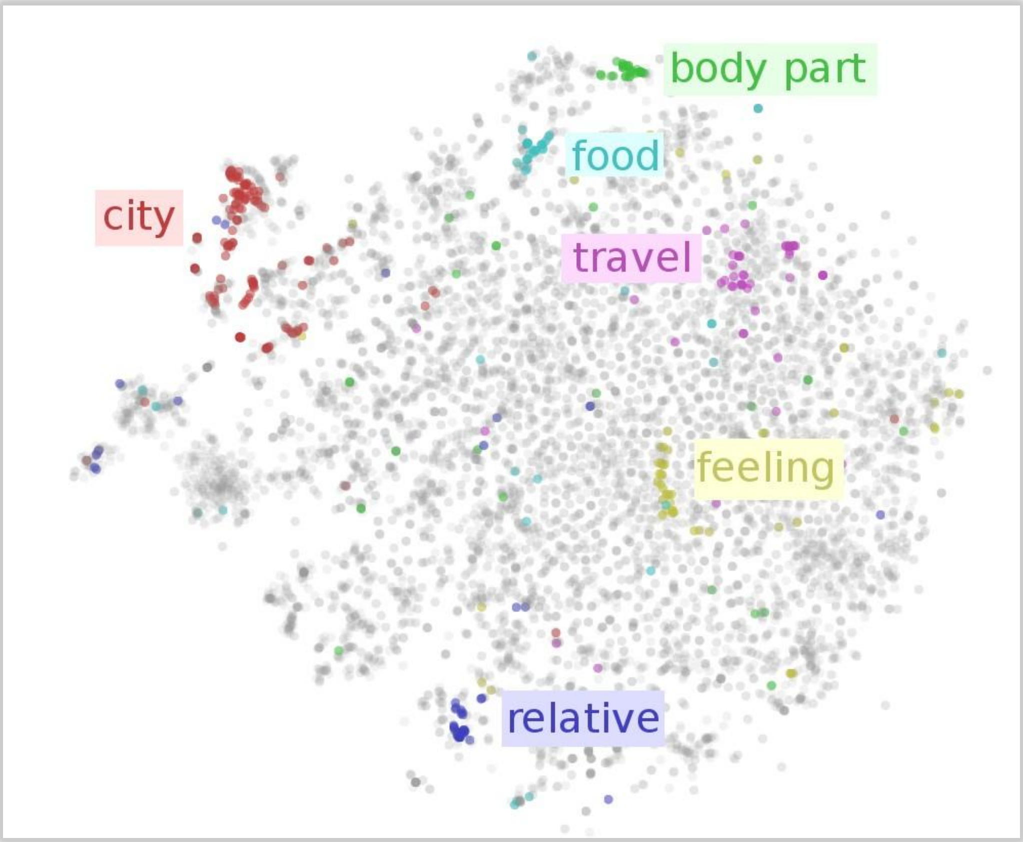
词语的表示学习-字词混合方法

- 词语由字或字符构成，一方面词语作为不可分割的单元可以获得一个表示；另一方面词语作为字的组合，通过字的表示也可以获得一个表示；两种表示结合得到更优表示。





词语的表示学习-示例



在低维、稠密的实数向量空间中，相似的词聚集在一起，在相同的历史上下文中具有相似的概率分布！

表示学习模型

- 词语的表示学习
- 短语的表示学习
- 句子的表示学习
- 文档的表示学习

短语的表示学习-词袋方法

- 假设短语由i个词语构成

$$ph_i = w_1 w_2 \cdots w_i$$

- 视短语为词袋，其表示为词语向量的平均

$$e(ph_i) = \frac{1}{i}\sum_{k=1}^{i} e(w_k)$$

猫吃鱼

vs.

鱼吃猫

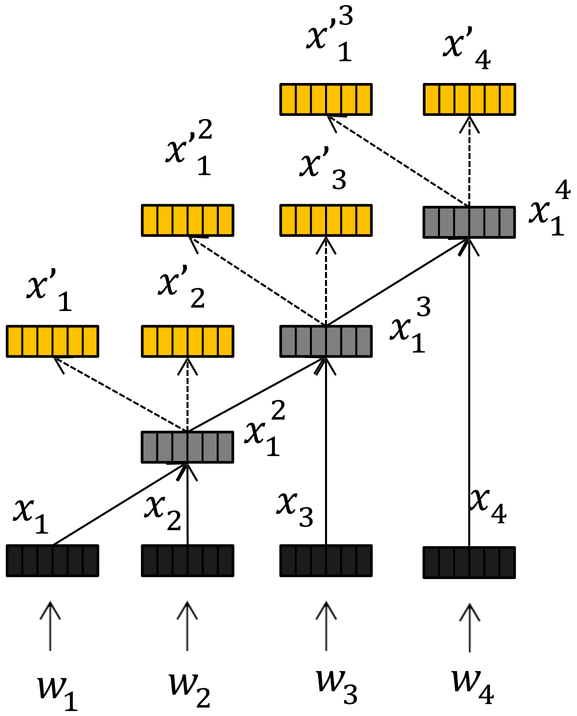
- 视短语为词袋，其表示为词语向量的加权平均

$$e(ph_i) = \sum_{k=1}^{i} v_k \cdot e(w_k)$$

短语的表示学习-递归自动编码器

- 假设短语由i个词语构成 $ph_i = w_1 w_2 \cdots w_i$

$i = 4$



$$x_1^2 = f\left(W^{(1)}[x_1 : x_2] + b^{(1)}\right)$$

$$[x'_1 : x'_2] = f\left(W^{(2)} x_1^2 + b^{(2)}\right)$$

$$E_{rec}[x_1 : x_2] = \frac{1}{2} \|[x_1 : x_2] - [x'_1 : x'_2]\|^2$$

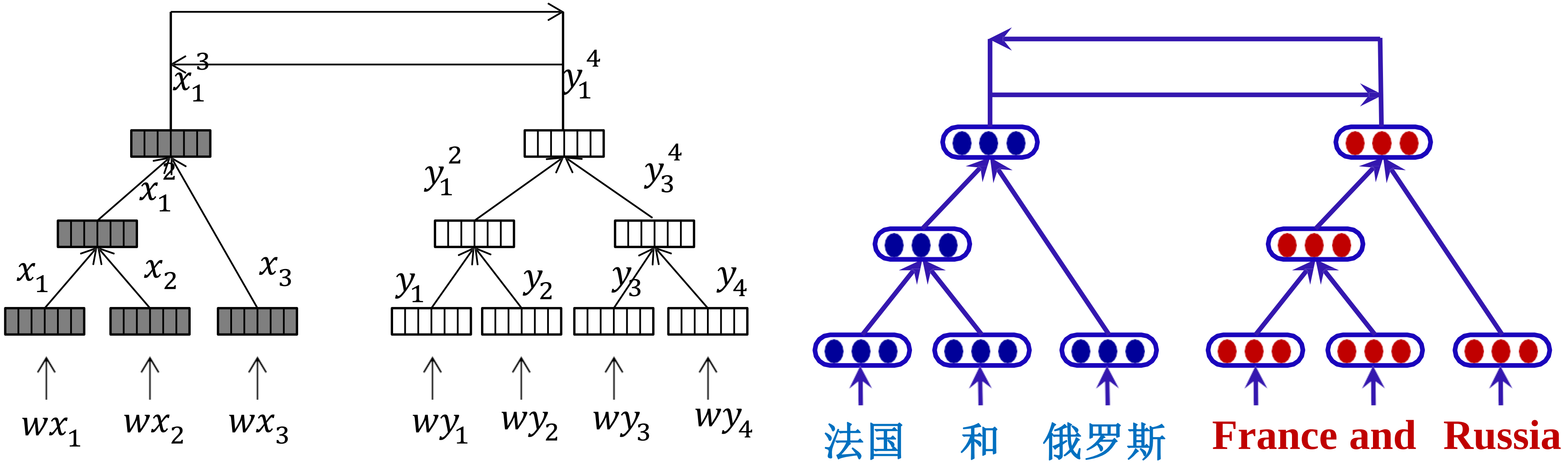
$$E_\theta(ph_i) = \operatorname*{argmin}_{bt \in A(ph_i)} \sum_{nd \in bt} E_{rec}(nd)$$

二叉树集合 二叉树中的内部节点

短语的表示学习-双语约束模型

- 假设已知(ph_x, ph_y)是两种语言互为翻译的短语，那么理论上这两个短语的语义表示应该是一样的

$$\underset{\theta}{\operatorname{argmin}}(f(x_1^3, y_1^4); \theta)$$



$$E(ph_x, ph_y; \theta) = \alpha E_{rec}(ph_x, ph_y; \theta) + (1-\alpha)\alpha E_{sem}(ph_x, ph_y; \theta)$$

重构误差

语义误差

短语的表示学习-双语约束模型

- 目标函数

正则化项

$$J = E\left(ph_x, ph_y; \theta\right) + \frac{1}{2}\lambda\|\theta\|^2$$

- 重构误差

$$E_{rec}\left(ph_x, ph_y; \theta\right) = E_{rec}(ph_x; \theta) + E_{rec}\left(ph_y; \theta\right)$$

- 语义误差

$$E_{sem}\left(ph_x, ph_y; \theta\right) = E_{sem}\left(ph_x|ph_y; \theta\right) + E_{sem}\left(ph_y|ph_x; \theta\right)$$

$$E_{sem}\left(ph_x|ph_y; \theta\right) = \frac{1}{2}\left\|f\left(e\left(ph_y\right)\right) - e(ph_x)\right\|^2$$

短语的表示学习-双语约束模型

新输入短语	RAE	BRAE
military force	core force main force labor force	military power military strength armed forces
at a meeting	to a meeting at a rate a meeting ,	at the meeting during the meeting at the conference
do not agree	one can accept i can understand do not want	do not favor will not compromise not to approve
each people in this nation	each country regards each country has its each other , and	every citizen in this country all the people in the country people all over the country

表示学习模型

- 词语的表示学习
- 短语的表示学习
- 句子的表示学习
- 文档的表示学习

句子的表示学习-词袋方法

- 假设句子由n个词语构成

$$s = w_1 w_2 \cdots w_n$$

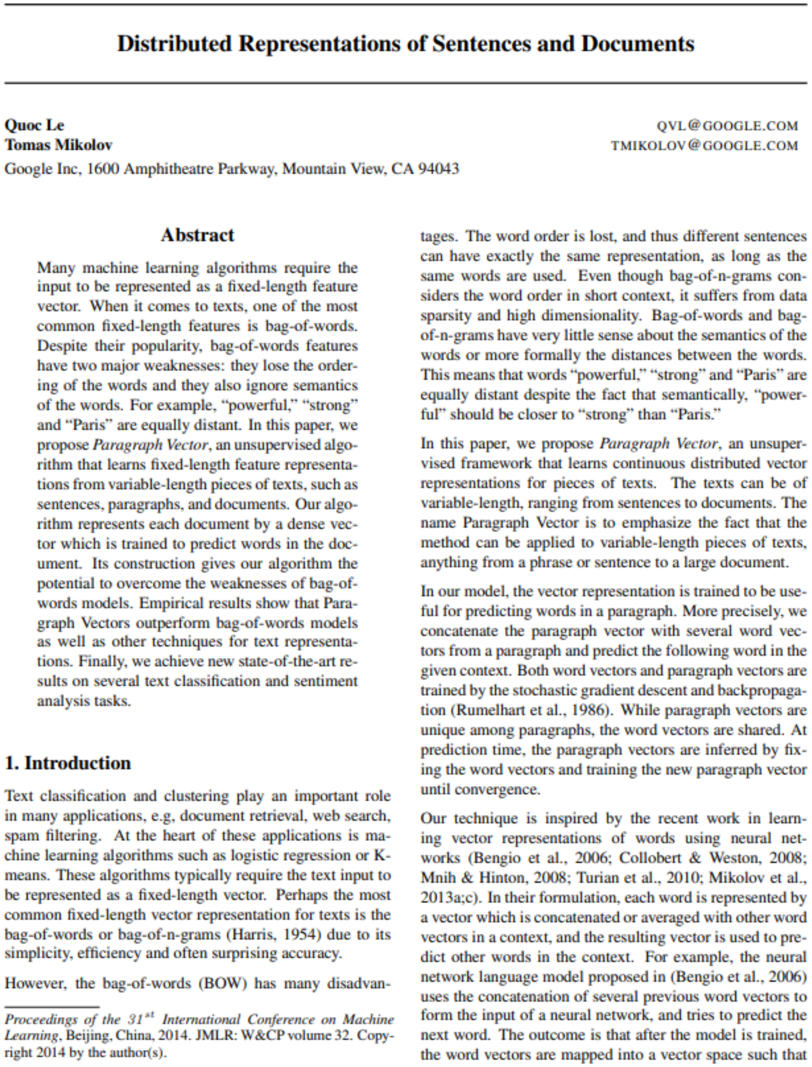
- 视句子为词袋，其表示为词语向量的平均

$$e(s) = \frac{1}{n} \sum_{k=1}^{n} e(w_k)$$

- 视句子为词袋，其表示为词语向量的加权平均

$$e(s) = \sum_{k=1}^{n} v_k \cdot e(w_k)$$

句子的表示学习-词袋方法



Distributed Representations of Sentences and Documents

Quoc Le QVL@GOOGLE.COM
Tomas Mikolov TMIKOLOV@GOOGLE.COM
Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

Abstract

Many machine learning algorithms require the input to be represented as a fixed-length feature vector. When it comes to texts, one of the most common fixed-length features is bag-of-words. Despite their popularity, bag-of-words features have two major weaknesses: they lose the ordering of the words and they also ignore semantics of the words. For example, "powerful," "strong" and "Paris" are equally distant. In this paper, we propose *Paragraph Vector*, an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents. Our algorithm represents each document by a dense vector which is trained to predict words in the document. Its construction gives our algorithm the potential to overcome the weaknesses of bag-of-words models. Empirical results show that Paragraph Vectors outperform bag-of-words models as well as other techniques for text representations. Finally, we achieve new state-of-the-art results on several text classification and sentiment analysis tasks.

1. Introduction

Text classification and clustering play an important role in many applications, e.g, document retrieval, web search, spam filtering. At the heart of these applications is machine learning algorithms such as logistic regression or K-means. These algorithms typically require the text input to be represented as a fixed-length vector. Perhaps the most common fixed-length vector representation for texts is the bag-of-words or bag-of-n-grams (Harris, 1954) due to its simplicity, efficiency and often surprising accuracy.

However, the bag-of-words (BOW) has many disadvantages. The word order is lost, and thus different sentences can have exactly the same representation, as long as the same words are used. Even though bag-of-n-grams considers the word order in short context, it suffers from data sparsity and high dimensionality. Bag-of-words and bag-of-n-grams have very little sense about the semantics of the words or more formally the distances between the words. This means that words "powerful," "strong" and "Paris" are equally distant despite the fact that semantically, "powerful" should be closer to "strong" than "Paris."

In this paper, we propose *Paragraph Vector*, an unsupervised framework that learns continuous distributed vector representations for pieces of texts. The texts can be of variable-length, ranging from sentences to documents. The name Paragraph Vector is to emphasize the fact that the method can be applied to variable-length pieces of texts, anything from a phrase or sentence to a large document.

In our model, the vector representation is trained to be useful for predicting words in a paragraph. More precisely, we concatenate the paragraph vector with several word vectors from a paragraph and predict the following word in the given context. Both word vectors and paragraph vectors are trained by the stochastic gradient descent and backpropagation (Rumelhart et al., 1986). While paragraph vectors are unique among paragraphs, the word vectors are shared. At prediction time, the paragraph vectors are inferred by fixing the word vectors and training the new paragraph vector until convergence.

Our technique is inspired by the recent work in learning vector representations of words using neural networks (Bengio et al., 2006; Collobert & Weston, 2008; Mnih & Hinton, 2008; Turian et al., 2010; Mikolov et al., 2013a;c). In their formulation, each word is represented by a vector which is concatenated or averaged with other word vectors in a context, and the resulting vector is used to predict other words in the context. For example, the neural network language model proposed in (Bengio et al., 2006) uses the concatenation of several previous word vectors to form the input of a neural network, and tries to predict the next word. The outcome is that after the model is trained, the word vectors are mapped into a vector space such that

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

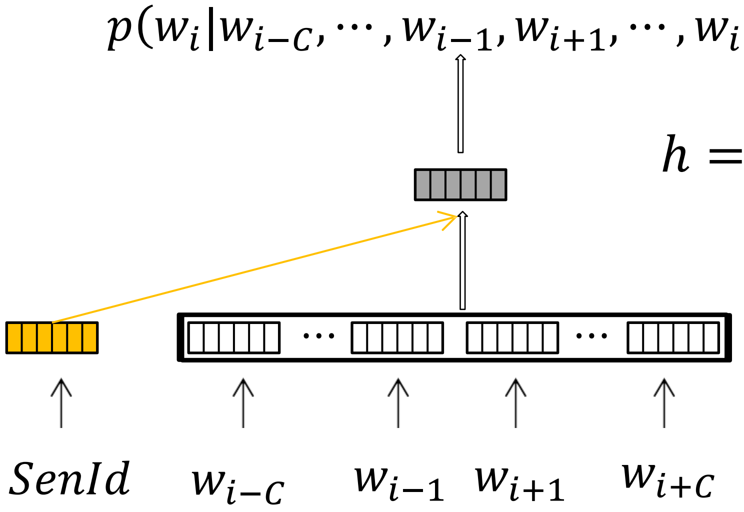
句子的表示学习-PV-DM模型

- 对于语料D中的M个句子，按照顺序，每个句子s_i对应一个序号i，该序号i可唯一代表这个句子。假设我们希望句子向量的维度为p，那么训练集中所有句子的向量对应一个矩阵$PV \in \mathcal{R}^{M \times p}$。序号为$i$的句子对应的向量是$PV$中的第$i$行 。

$$
\begin{gathered}
s_1 = w_1^1 w_2^1 \cdots w_{n_1}^1 \\
s_2 = w_1^2 w_2^2 \cdots w_{n_2}^2 \\
\vdots \\
s_i = w_1^i w_2^i \cdots w_{n_i}^i \\
\vdots \\
s_M = w_1^M w_2^M \cdots w_{n_M}^M
\end{gathered}
$$

句子的表示学习-PV-DM模型

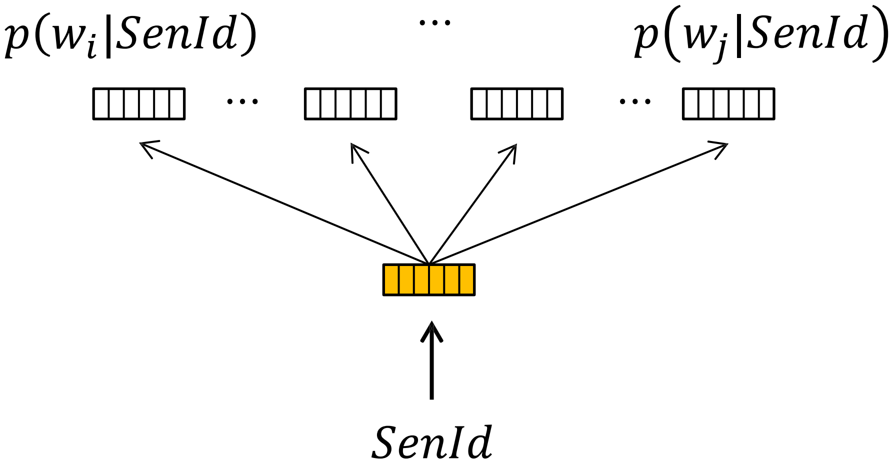
- PV-DM模型（Paragraph Vector with sentence as Distributed Memory）是CBOW 的扩展，将上下文所在的句子视为一个记忆单元，对于任意一个n-元组$(w_i, C) = w_{i-C}\cdots w_{i-1}w_iw_{i+1}\cdots w_{i+C}$以及该$n$-元组所在的句子序号$SenId$，我们将$SenId$和$WC = w_{i-C}\cdots w_{i-1}w_{i+1}\cdots w_{i+C}$作为输入，计算句子和上下文词语的平均词向量（或采用向量拼接的方式）


$p(w_i|w_{i-C},\cdots,w_{i-1},w_{i+1},\cdots,w_{i+C},SenId)$
$\cdots$
$\cdots$
$SenId$
w_{i-C}
w_{i-1}
w_{i+1}
w_{i+C}

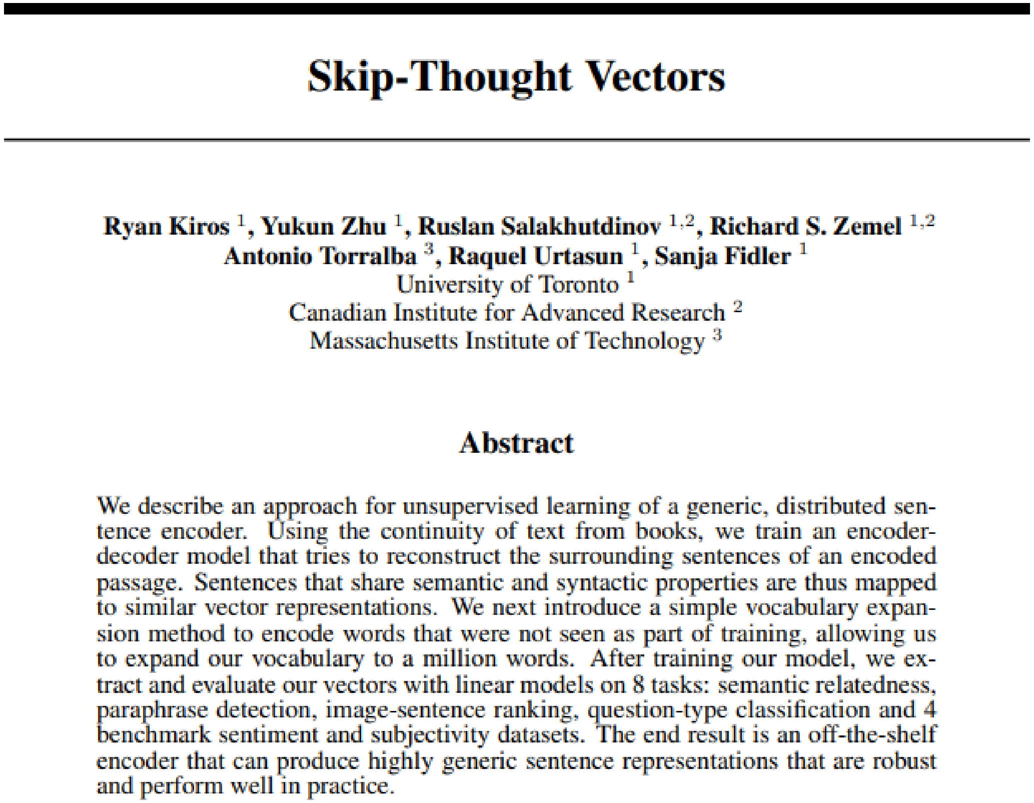
$$h = \frac{1}{2C+1}\left(e(SenId) + \sum_{i-C\le k\le i+C, k\neq i} e(e_k)\right)$$

句子的表示学习-PV-DBOW模型

- 对Skip-gram模型的扩展，形成了句子表示模型PV-DBOW（Distributed Bag-of-Words version of Paragraph Vector）。该模型以句子为输入，以句子中随机抽样的词语为输出，即要求句子能够预测句中的任意词语。其目标函数设计和训练方式与Skip-gram模型相同



句子的表示学习-Skip-Thought模型



Skip-Thought Vectors

Ryan Kiros [1], **Yukun Zhu** [1], **Ruslan Salakhutdinov** [1,2], **Richard S. Zemel** [1,2]
Antonio Torralba [3], **Raquel Urtasun** [1], **Sanja Fidler** [1]
University of Toronto [1]
Canadian Institute for Advanced Research [2]
Massachusetts Institute of Technology [3]

Abstract

We describe an approach for unsupervised learning of a generic, distributed sentence encoder. Using the continuity of text from books, we train an encoder-decoder model that tries to reconstruct the surrounding sentences of an encoded passage. Sentences that share semantic and syntactic properties are thus mapped to similar vector representations. We next introduce a simple vocabulary expansion method to encode words that were not seen as part of training, allowing us to expand our vocabulary to a million words. After training our model, we extract and evaluate our vectors with linear models on 8 tasks: semantic relatedness, paraphrase detection, image-sentence ranking, question-type classification and 4 benchmark sentiment and subjectivity datasets. The end result is an off-the-shelf encoder that can produce highly generic sentence representations that are robust and perform well in practice.

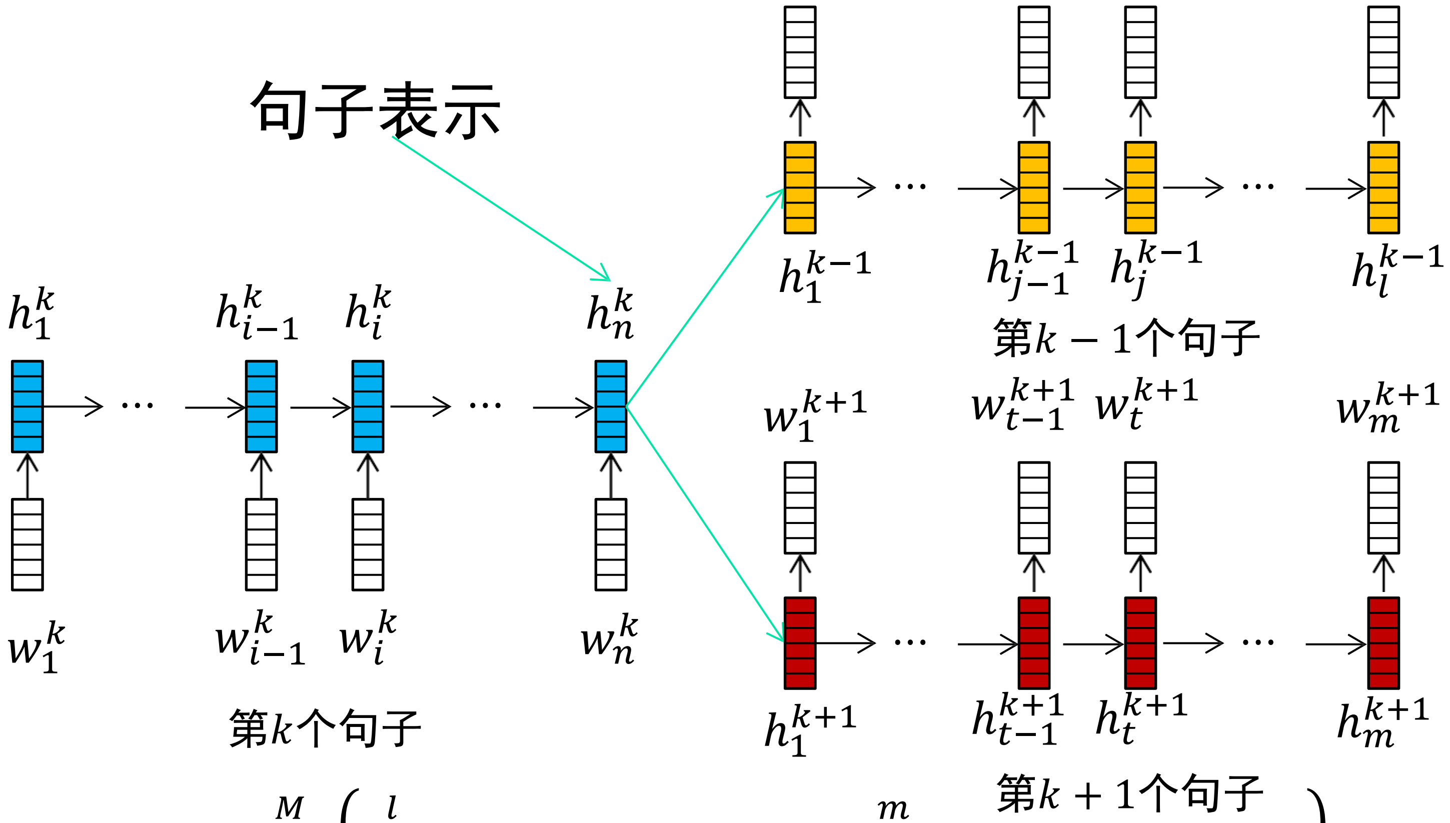
Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, Sanja Fidler: **Skip-Thought Vectors.** NIPS 2015: 3294-3302

句子的表示学习-Skip-Thought模型

- Skip-Thought句子表示方法类似于PV-DBOW模型，不同于PV-DBOW模型利用句子预测句中的词语，Skip-Thought模型利用当前句子s_k预测前一个句子s_{k-1}与后一个句子s_{k+1}。该模型认为，文本中连续出现的句子$s_{k-1}s_k s_{k+1}$表达的意思比较接近，因此，根据句子s_k的语义，可以重构出前后两个句子。

$$
\begin{gathered}
s_1 = w_1^1 w_2^1 \cdots w_{n_1}^1 \\
s_2 = w_1^2 w_2^2 \cdots w_{n_2}^2 \\
\vdots \\
s_k = w_1^k w_2^k \cdots w_{n_k}^k \\
\vdots \\
s_M = w_1^M w_2^M \cdots w_{n_M}^M
\end{gathered}
$$

句子的表示学习-Skip-Thought模型

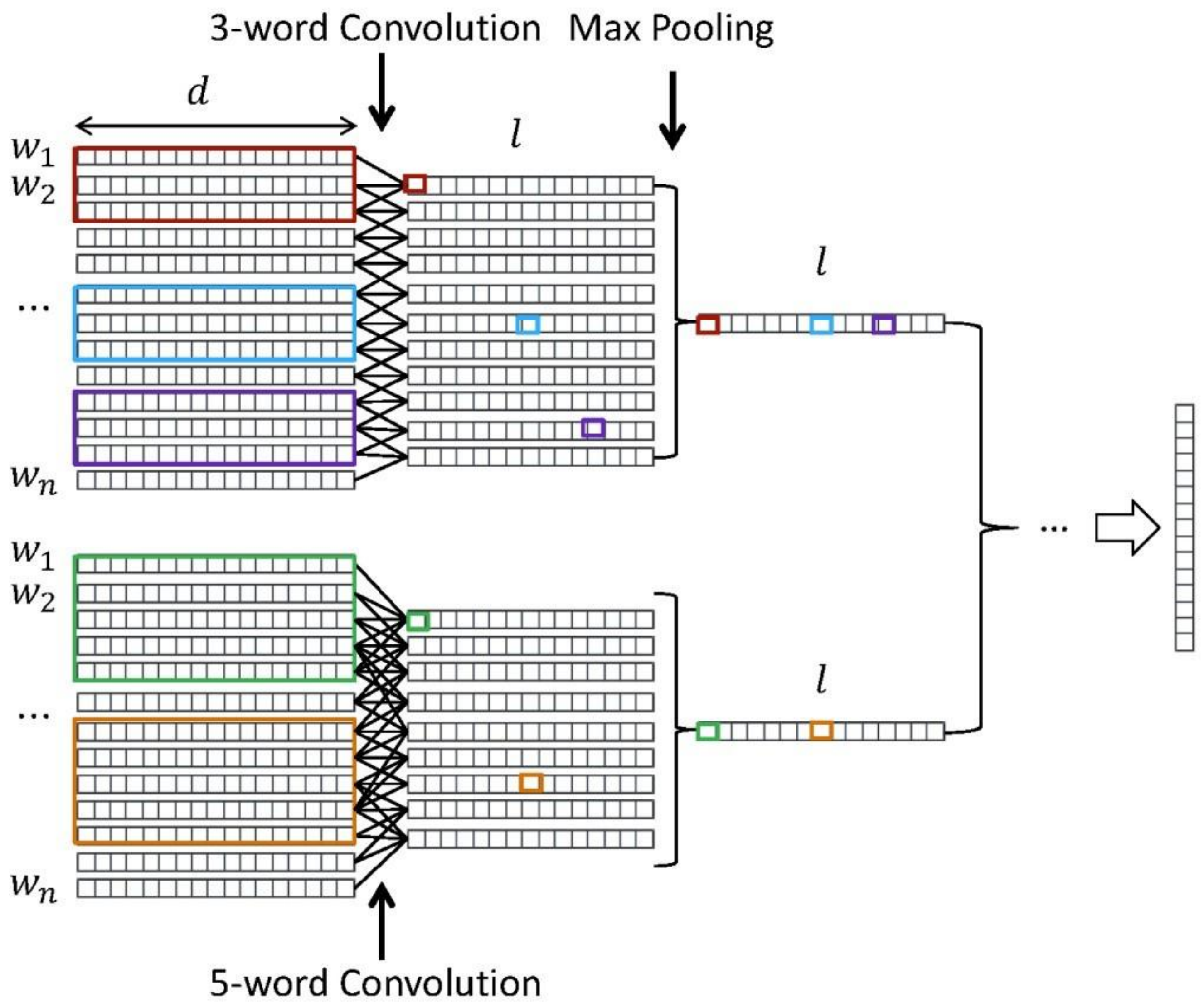


$$\sum_{k=1}^{M}\left\{\sum_{j=1}^{l} p\left(w_j^{k-1}|w_{<j}^{k-1},h_n^k\right)+\sum_{t=1}^{m} p\left(w_t^{k-1}|w_{<t}^{k-1},h_n^k\right)\right\}$$

句子的表示学习-CNN模型

- 对于一个句子，卷积神经网络CNN以每个词的词向量为输入，通过顺序地对上下文窗口进行卷积（Convolution）总结局部信息，并利用池化层（Pooling）提取全局的重要信息，再经过其他网络层（卷积池化层、Dropout层、线性层等），得到固定维度的句子向量表达，以刻画句子全局性的语义信息。

句子的表示学习-CNN模型



句子的表示学习-CNN模型

- 给定n个词语的句子$w_1 w_2 \cdots w_n$，每个词语首先利用预训练或随机初始化的词向量矩阵$L \in \mathcal{R}^{|V| \times d}$映射为词向量列表$X = [x_1, x_2, \cdots, x_n]$。对于任意一个$h$长度的窗口$x_{i:i+h-1}$，卷积层采用卷积算子$F_t$（$1 \le t \le l$，$l$表示卷积算子数目），得到一个局部特征$y_i^t$：

$$y_i^t = F_t(W x_{i:i+h-1} + b)$$

- 卷积算子F_t从$x_{1:h-1}$到$x_{n-h+1:n}$遍历整个句子，得到特征列表：$y^t = [y_1^t, y_2^t, \cdots, y_{n-h+1}^t]$。为了将不定长的$y^t$转换为定长输出，池化成为不可或缺的操作。最大池化是最流行的池化方法，认为$\hat{y}^t = \max y^t$代表了卷积算子F_t在整个句子上获得的最重要特征。l个卷积算子将得到一个l-维的特征向量$y = [\hat{y}^1, \hat{y}^2, \cdots, \hat{y}^l]$。

表示学习模型

- 词语的表示学习
- 短语的表示学习
- 句子的表示学习
- 文档的表示学习

文档的表示学习-词袋方法

- 假设文档由n个词语构成

$$s = w_1 w_2 \cdots w_n$$

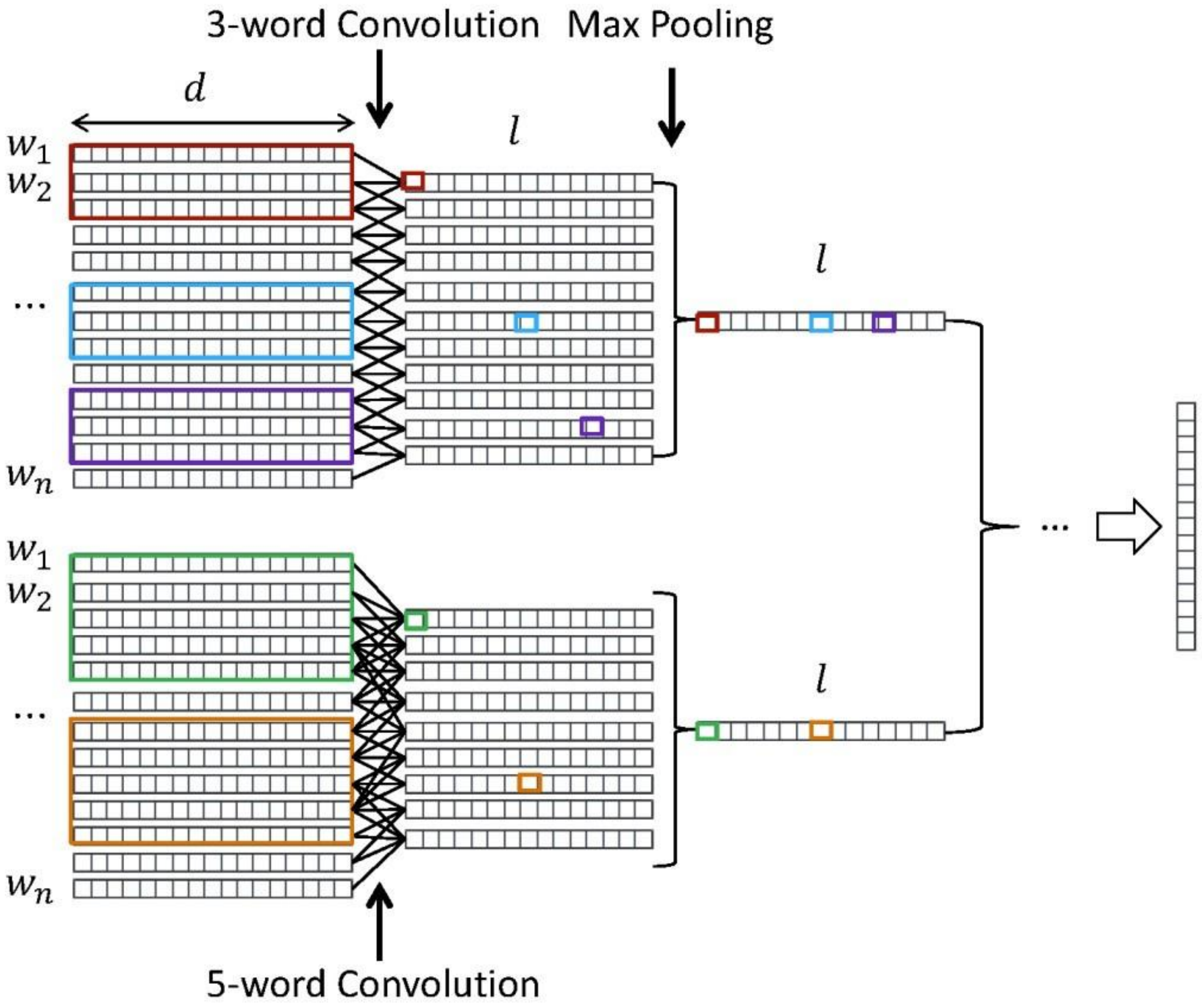
- 视文档为词袋，其表示为词语向量的平均

$$e(s) = \frac{1}{n} \sum_{k=1}^{n} e(w_k)$$

- 视文档为词袋，其表示为词语向量的加权平均

$$e(s) = \sum_{k=1}^{n} v_k \cdot e(w_k)$$

文档的表示学习-词袋方法



文档的表示学习-层次化模型

$$x = f(We_D + b) \in R^C \quad p_k = \frac{exp(x_k)}{\sum_{k'=1}^{C} exp(x_{k'})}$$

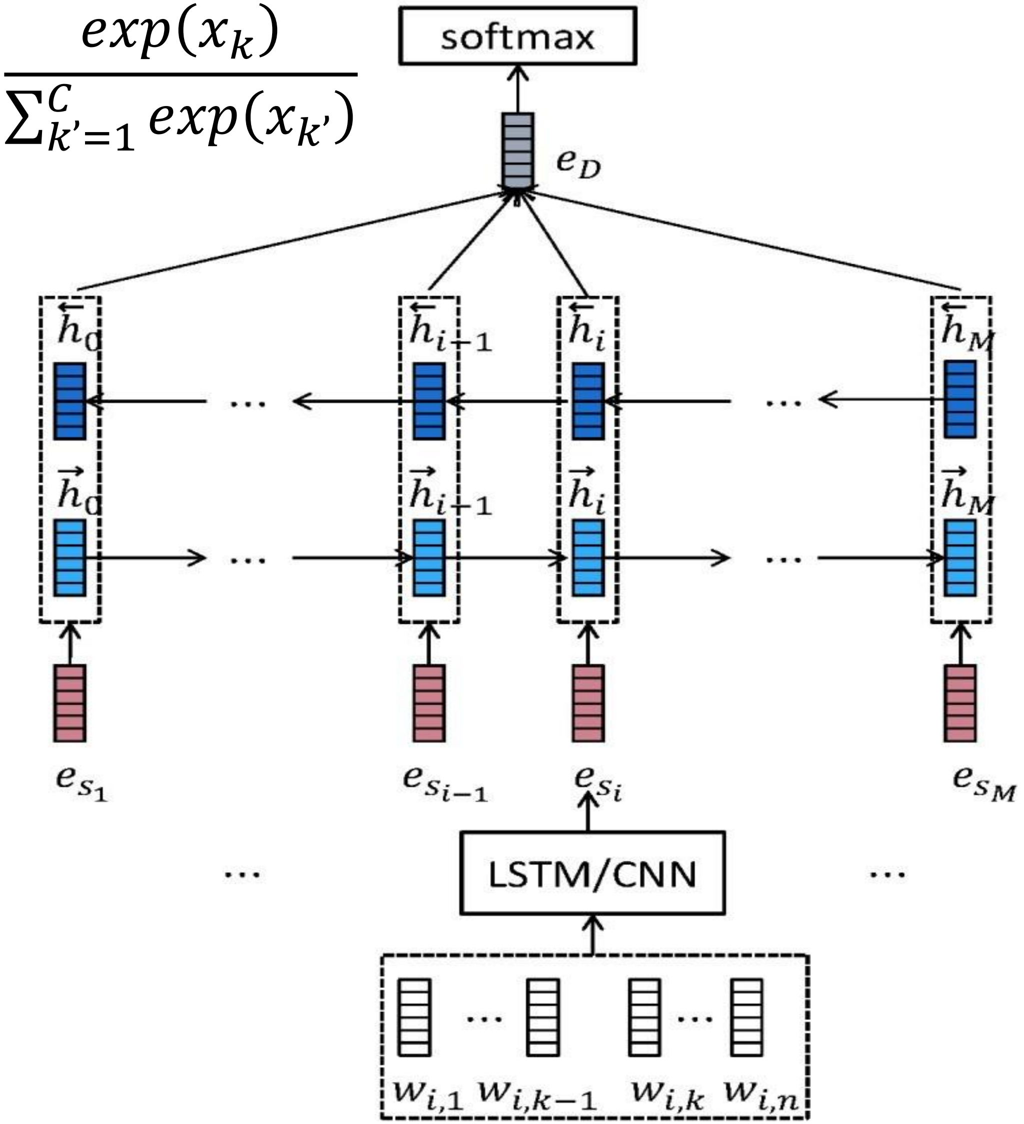
$$e_D = \sum_{i=1}^{M} v_i h_i$$

$$\overrightarrow{h_i} = LSTM\left(e_{s_i}, \overrightarrow{h_{i-1}}\right)$$

$$\overleftarrow{h_i} = LSTM\left(e_{s_i}, \overleftarrow{h_{i+1}}\right)$$

$$e_{s_i} = LSTM\left(w_{i,1} \cdots w_{i,n}\right)$$

$$e_{s_i} = CNN\left(w_{i,1} \cdots w_{i,n}\right)$$



本部分小结

- 向量空间模型
  特征项 vs. 特征项权重
- 表示学习模型：
  - 词语的表示学习
  - 短语的表示学习
  - 句子的表示学习
  - 文档的表示学习

开源工具

1. Google Word2Vec, http://code.google.com/p/word2vec/

2. GloVe, https://github.com/stanfordnlp/GloVe

3. Skip-Thought, https://github.com/ryankiros/skip-thoughts

4. FastText, https://github.com/facebookresearch/fastText

… …

欢迎提问！