



Mobile
Innovation
Lab

Word Embeddings

Jéssica Rodrigues da Silva

AI Engineer at Samsung P&D

PATROCÍNIO:



INFOMACH
TECNOLOGIA PARA NEGÓCIOS

Saúdemobi

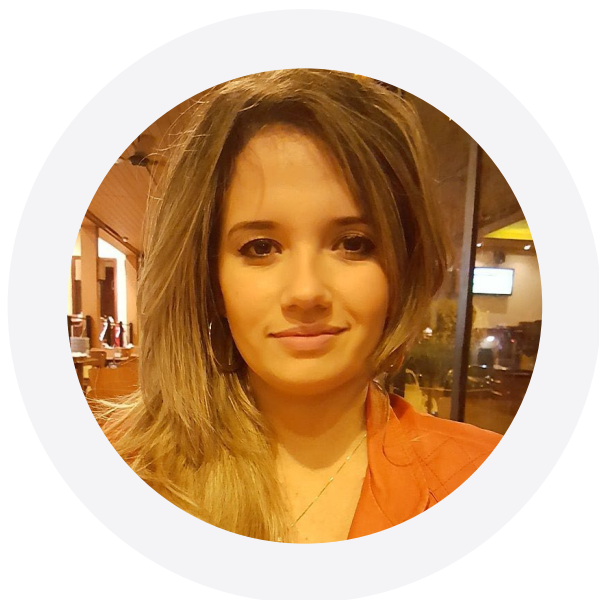


www.deeplearning.com.br

APOIO:



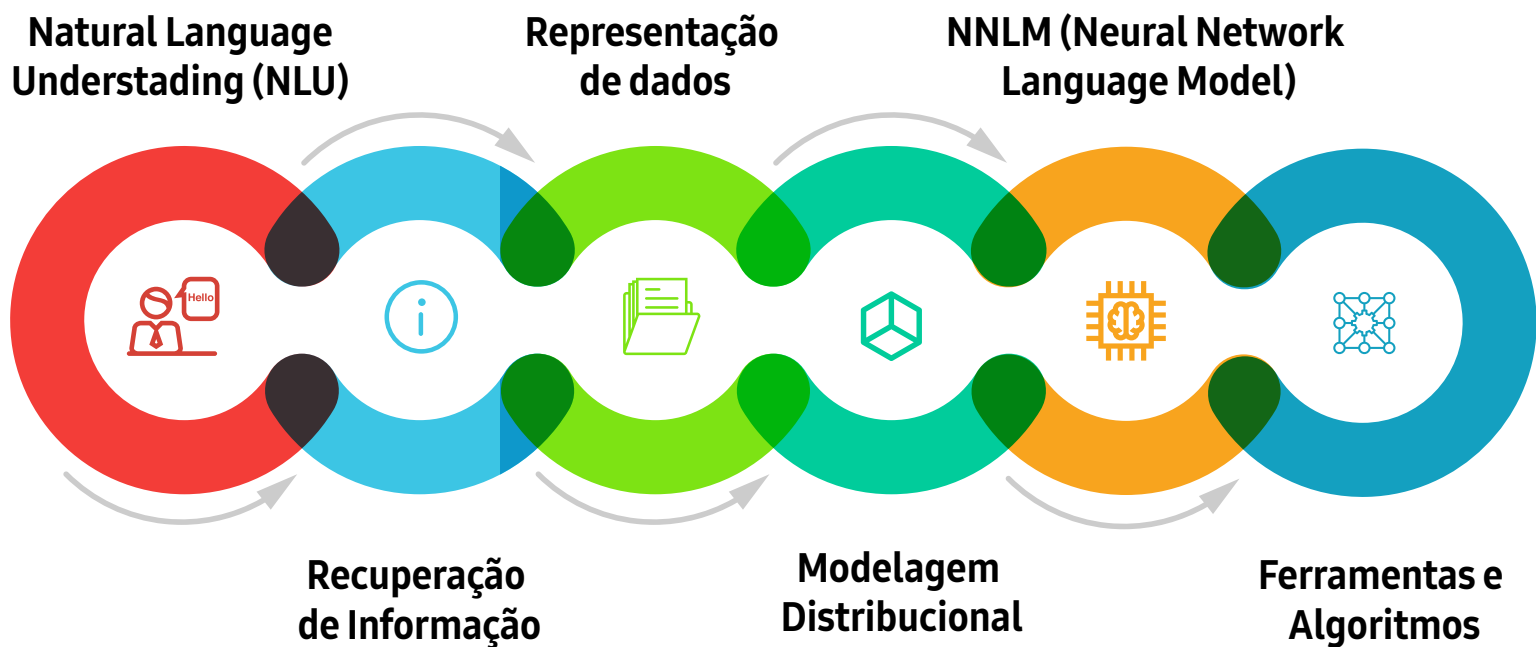
DATA^H
01



Jéssica Rodrigues da Silva

- Mestranda em Ciência da Computação - Natural Language *Processing and Machine Learning* - pela UFSCar – Universidade Federal de São Carlos;
- Graduada em Ciência da Computação;
- AI Engineer na Samsung Instituto de P&D da Amazônia;
- LinkedIn: in/jessica-rodrigues-silva

O Processamento da Língua






Natural Language Understanding (NLU)

Como fazer com que a máquina compreenda o que quero dizer?


Ex:



Todas Notícias Vídeos Shopping Imagens

Aproximadamente 848.000 resultados (0,47 segundos)


Principais notícias



Como será o julgamento de Lula no TRF-4, em Porto Alegre

IstoÉ Dinheiro

4 horas atrás




Entenda como será o julgamento de Lula nesta quarta-feira

Istoé

8 horas atrás

→ Mais sobre quando será o julgamento do Lula



Todas Notícias Vídeos Shopping Imagens Mais Configurações Ferramentas

Aproximadamente 1.080.000 resultados (0,44 segundos)

Eleições no Brasil – Wikipédia, a enciclopédia livre

https://pt.wikipedia.org/wiki/Eleicoes_no_Brasil

As eleições no Brasil acontecem a cada dois anos, a exemplo dos presidentes, governadores, deputados e senadores em 2014 e dos prefeitos e vereadores em 2016. Os mandatos de vereadores, prefeitos, deputados estaduais, federais, governadores e do presidente da República duram quatro anos; o dos senadores ...

História · Sistema eleitoral · Infraestrutura e processo · Direitos políticos

Presidente da República: como é eleito e quem pode concorrer ...

www.politize.com.br/presidente-da-republica-como-e-eleito/

20 de jun de 2017 - Como é eleito o Presidente da República? Entenda as ... Assim como na maior parte dos sistemas presidencialistas do mundo, o presidente brasileiro é eleito pelo sistema majoritário. No nosso ... Ela também pode ocorrer em casos de plebiscitos ou referendos nacionais – que ocorrem muito raramente.

Eleições no Brasil - InfoEscola

<https://www.infoescola.com/direito/eleicoes-no-brasil/>

As eleições ocorrem no primeiro domingo de outubro. Os cargos correspondentes ao Poder Legislativo (Senadores, Deputados Federais, Deputados Estaduais e Vereadores) são disputados em turno único. Para os cargos do Poder Executivo (Presidente, Governadores e Prefeitos), pode haver segunda turno.



Natural Language Understanding (NLU)

Ex:

52% 22:20

jornal.usp.br > universidade > amp

O Instituto de **Ciências Matemáticas e de Computação (ICMC)** da **USP**, em **São Carlos**, está com **inscrições** ... Vale...

O li
de
Car

quando abrem as inscrições para o doutorado em Ciência da Computação da UFG



Vamos ver

Doutorado em Ciência da Computação UFG e UFMS | Institu...
www.inf.ufg.br > doutorado

Divulgação do resultado de seleção Alunos Especiais - **Doutorado** 2017/2 · Aula Inaugural da Pós-Graduação INF/...

Do
Co
ww

O n
Ciê
Oe:

Pesquisa

O que você pode fazer?



Como fazer com que a máquina compreenda o que quero dizer?

Ex:

51% 22:21

Qual o cenário da febre amarela em 2017. 12/01/2018 17h29...

ain

Qual o cenário da febre amarela no Brasil

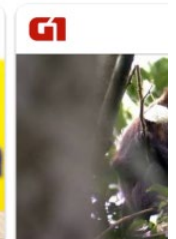


Aqui estão as últimas



Uruguai faz vacinação fracionada contra febre amarela para pessoas que...

AMP - 1 hora atrás



Febre amarela Paulo virou ár...

AMP - 8 dias atrás

Pesquisa

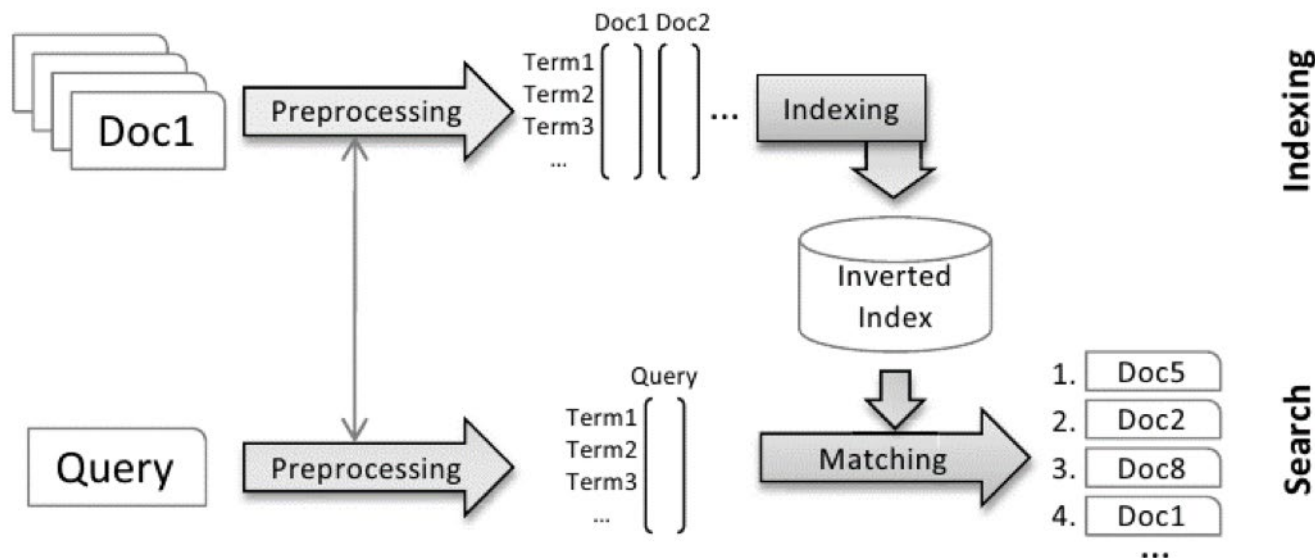
Notícias sobre São Paulo





Information Retrieval (IR)

- Algoritmos tradicionais de IR lidam diretamente com o token (palavra);
- Línguas naturais são esparsas e ambíguas.





Representação de dados

- Representação de dados x Modelos de Aprendizagem;
- Símbolos (palavra, pixel, onda) só devem existir na entrada e saída de uma rede de neurônios;
- Dentro de nossos cérebros são apenas grandes vetores de atividade;
- Essas representações são chamadas de vetores de pensamento, ou ***thought vectors***.



Representação de dados

“

*If we can read every English document on the web, and turn each sentence into a **thought vector**, you've got plenty of data for training a system that can reason like people do.*

*What I think is going to happen over the next few years is this ability to turn sentences into **thought vectors** is going to rapidly change the level at which we can understand documents.*

Geoffrey Hinton

”



Representação de dados

Vídeo



to the 14 of these weights that's 100



Modelagem Distribucional

“

YOU SHALL KNOW A WORD BY THE COMPANY IT KEEPS!

John R. Firth (1957)

”



Modelagem Distribucional

Os métodos distributivos assumem que o sentido de uma palavra está relacionado à distribuição de palavras em torno dela, suposição que está baseada na hipótese distribucional de (HARRIS, 1954 apud ZANZOTTO et al., 2010).

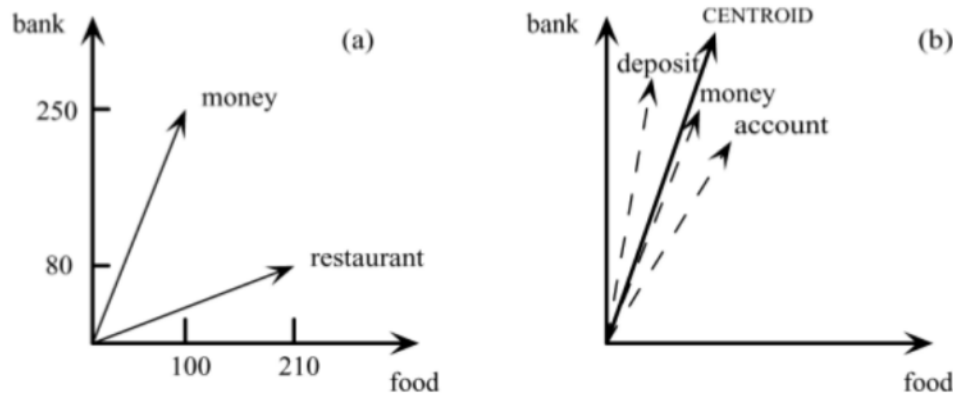
- A garrafa de vinho está na mesa.
 - Todo mundo gosta de vinho.
 - Vinho faz você ficar bêbado.
 - Nós fazemos vinho de uva.
-
- Cluster 1:
 - O **banco** funciona 24 horas.
 - O **banco** aprovou meu financiamento.
 - Cluster 2:
 - O **banco** está com déficit de sangue.
 - Deve-se armazenar esses dados em um **banco**.



Modelagem Distribucional

	arts	boil	data	function	large	sugar	summarized	water
apricot	0	1	0	0	1	1	0	1
pineapple	0	1	0	0	1	1	0	1
digital	0	0	1	1	1	0	1	0
information	0	0	1	1	1	0	1	0

Fonte: Retirado de (JURAFSKY; MARTIN, 2008)

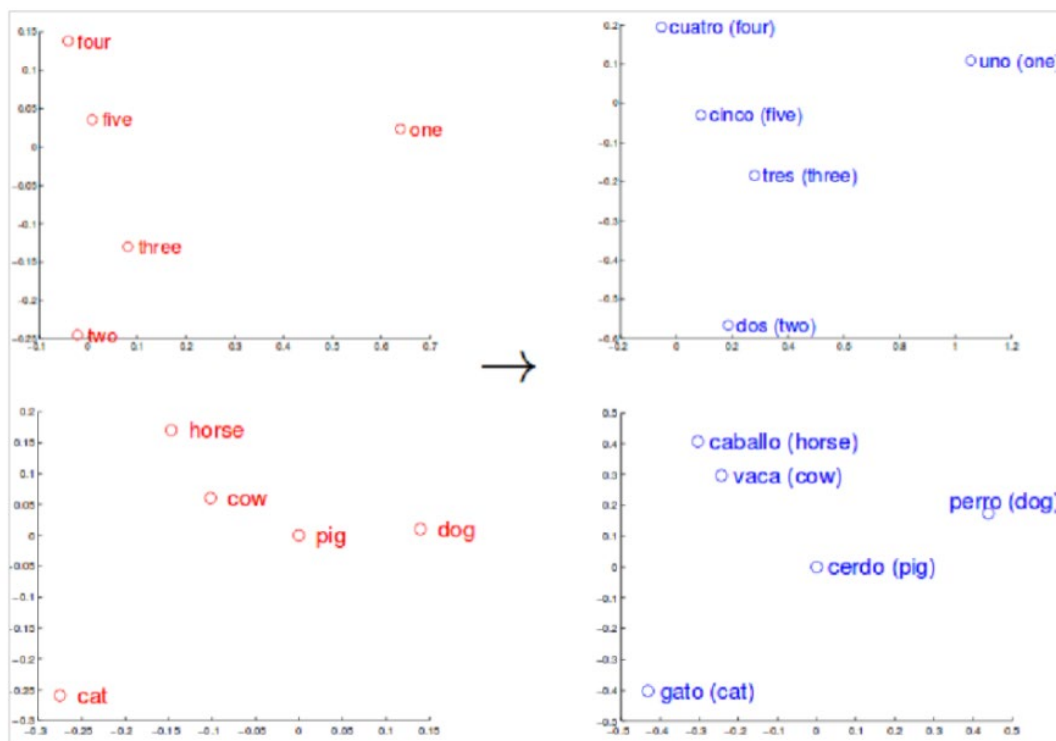


Fonte: Retirado de (JURAFSKY; MARTIN, 2008)



Modelagem Distribucional

O objetivo é gerar vetores contendo números de tal forma que palavras similares de acordo com seus contextos estarão “próximas” no espaço vetorial.



Fonte: Retirado de (MIKOLOV et al., 2013)



NNLM (Neural Network Language Model)

Proposta por Bengio et al., (2003), onde uma rede neural feedforward, com uma camada linear e uma camada escondida não-linear, é usada para aprender a representação vetorial de palavras e um modelo de linguagem estatístico.

Mikolov et al. (2009) propôs que os vetores de palavras fossem primeiro aprendidos usando uma rede neural com uma única camada oculta e depois usados para treinar o NNLM.

Mikolov et al. (2013) estende diretamente essa arquitetura, focando apenas no primeiro passo onde os vetores de palavras são aprendidos usando um modelo simples.



NNLM (Neural Network Language Model)

Proposta por Bengio et al., (2003), onde uma rede neural feedforward, com uma camada linear e uma camada escondida não-linear, é usada para aprender a representação vetorial de palavras e um modelo de linguagem estatístico.

Mikolov et al. (2009) propôs que os vetores de palavras fossem primeiro aprendidos usando uma rede neural com uma única camada oculta e depois usados para treinar o NNLM.

Mikolov et al. (2013) estende diretamente essa arquitetura, focando apenas no primeiro passo onde os vetores de palavras são aprendidos usando um modelo simples.



Ferramentas e Algoritmos

Word2Vec

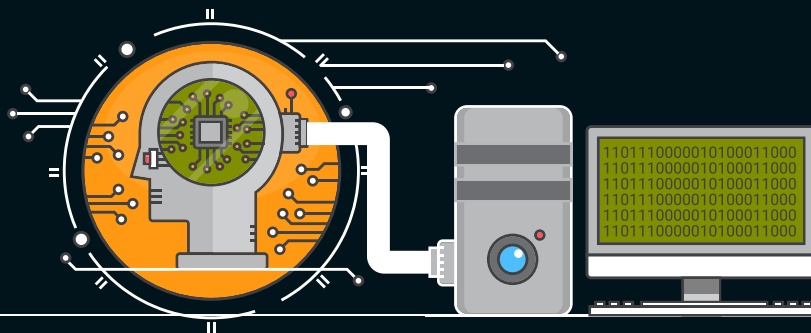
Doc2Vec

Wang2Vec

GloVe

FastText

Sense-Specific
Word Embeddings





Word2Vec

- Criado por Thomas Mikolov et al. em 2013;
- Eficiente computacionalmente;
- Algoritmos: Skip-gram e CBOW;
- Frameworks: Gensim, TensorFlow.

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.

1 Introduction

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

However, the simple techniques are at their limits in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited - the performance is usually dominated by the size of high quality transcribed speech data (often just millions of words). In

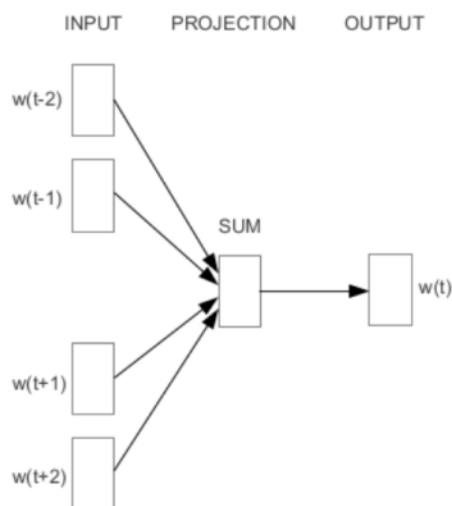


A similaridade das representações de palavras vai além das simples regularidades sintáticas. Ex:

$$\text{vetor(rei)} - \text{vetor(homem)} + \text{vetor(mulher)} = \text{vetor(rainha)}$$

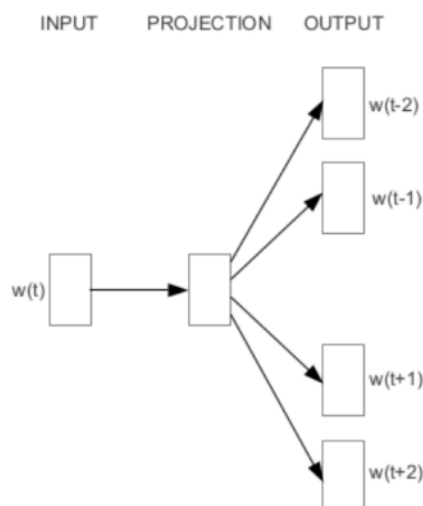
- **CBOW** (Continuous Bag of Words): camada escondida não-linear é removida e a camada de projeção é compartilhada para todas as palavras (não apenas a matriz de projeção). Essa arquitetura é chamada de modelo de saco de palavras (bag of words), pois a ordem das palavras não influencia a projeção.
- **Skip-Gram** - Tenta maximizar a classificação de uma palavra com base em outra da mesma sentença. Mais precisamente, usa-se cada palavra atual como uma entrada para um classificador log-linear para prever palavras dentro de um intervalo anterior e posterior à palavra atual.

Word2Vec



CBOW

Fonte: Retirado de (MIKOLOV et al., 2013)

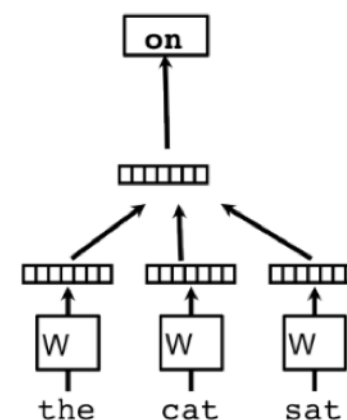


Skip-gram

Classifier

Average/Concatenate

Word Matrix



Fonte: Retirado de (LE; MIKOLOV, 2014)



Prever uma palavra dada as outras palavras em um contexto: Cada palavra é mapeada para um vetor, representado por uma coluna em uma matriz W . A coluna é indexada pela posição da palavra no vocabulário. A concatenação ou soma dos vetores é usada como feature para a predição da próxima palavra na sentença.

Mais formalmente, dada uma sequência de palavras $w_1, w_2, w_3, \dots, w_T$, o objetivo do modelo é maximizar a probabilidade de log média:

$$\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \dots, w_{t+k})$$

A tarefa de predição normalmente é feita através de um classificador multi-classe, como softmax.



- Criado por Thomas Mikolov e Le em 2014;
- Eficiente computacionalmente;
- Algoritmos: PV-DM e PV-DBOW;
- Frameworks: Gensim.

Distributed Representations of Sentences and Documents

Quoc Le
Tomas Mikolov

Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

QVL@GOOGLE.COM
TMIKOLOV@GOOGLE.COM

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 ma-

tages. 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.”

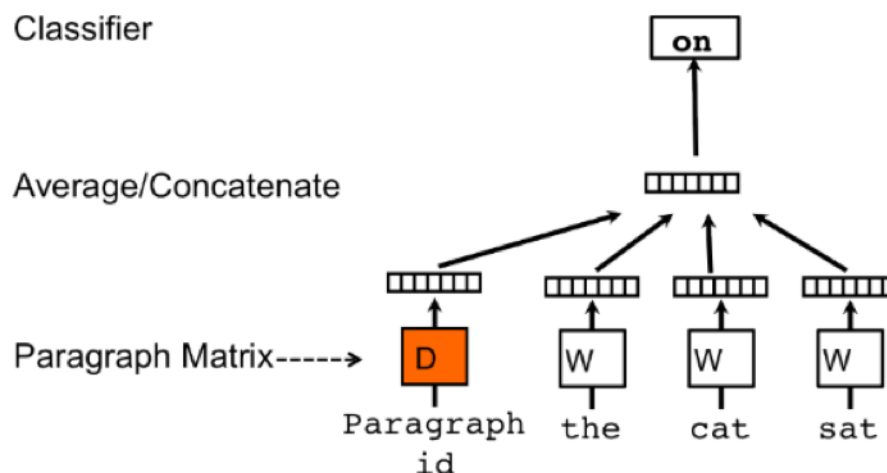
In this paper, we propose *Paragraph Vector*, an unsupervised framework that learns continuous distributed vector representations for pieces of texts. The texts can be of variable-length, ranging from sentences to documents. The name Paragraph Vector is to emphasize the fact that the method can be applied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



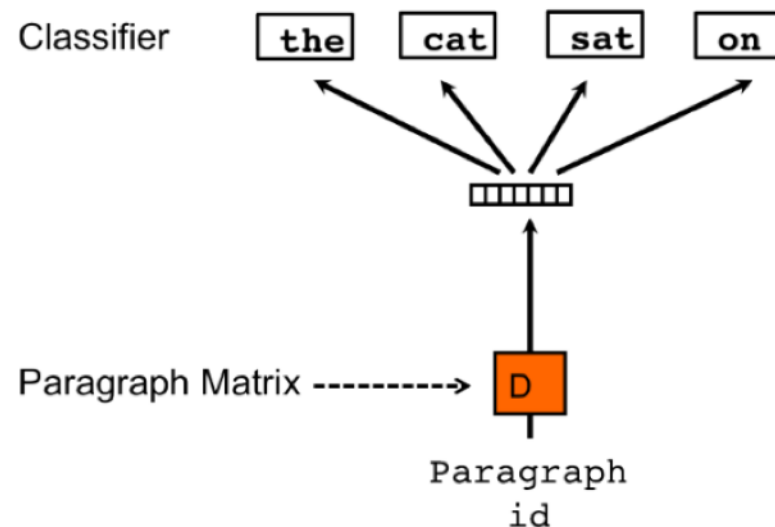
PV-DM (*Distributed Memory Model of Paragraph Vectors*): a concatenação ou média do vetor de parágrafo com o contexto de três palavras é usada para prever a quarta palavra. O vetor de parágrafo representa as informações que faltam no contexto atual e pode atuar como uma memória do tópico do parágrafo.



Fonte: Retirado de (LE; MIKOLOV, 2014)



PV-DBOW (*Distributed Bag of Words version of Paragraph Vector*): As palavras de contexto são ignoradas na entrada e previstas aleatoriamente na saída a partir do vetor do parágrafo.



Fonte: Retirado de (LE; MIKOLOV, 2014)



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

- Criado por Jeffrey Pennington, Richard Socher e Christopher Manning em 2014;
- Eficiente computacionalmente;
- Algoritmos: Algoritmo baseado no Skip-gram de Mikolov et al. (2013);
- Frameworks: Gensim

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 sub-structure, 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.

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 mod-

1 Introduction



GloVe é um módulo de regressão log-bilinear global que combina as vantagens de duas principais abordagens de modelos da literatura:

- Métodos de fatoração de matriz global, como o LSA;
- Métodos de janela de contexto local, como o Skip-gram de Mikolov et al. (2013);

Aproveita de forma eficiente as estatísticas de ocorrências de palavras. Essas estatísticas são calculadas considerando-se uma matriz de co-ocorrência **X** de palavra-palavra, com cada entrada **X_{ij}** representando o número de vezes que a palavra **j** ocorre no contexto da palavra **i**. A fórmula abaixo é a probabilidade da palavra **j** aparecer no contexto da palavra **i**.

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$$



Exemplo:

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

Fonte: Retirado de (PENNINGTON et al., 2014)

Dadas as palavras **i** (ice) e **j** (steam), a relação entre elas pode ser estudada através da proporção de suas probabilidades de co-ocorrência com várias palavras **k**.

Para palavras **k** relacionadas a ice mas não a steam, como é o caso de $k=solid$, espera-se um valor grande para P_{ik}/P_{jk} . De modo análogo, espera-se um valor pequeno para palavras relacionadas a steam e não a ice, como $k=gas$.



- Criado por Piotr Bojanowski, Edouard Grave, Armand Joulin e Tomas Mikolov em 2016;
- Eficiente computacionalmente;
- Algoritmos: Algoritmo baseado no Skip-gram de Mikolov et al. (2013);
- Frameworks: Gensim

Enriching Word Vectors with Subword Information

Piotr Bojanowski* and Edouard Grave* and Armand Joulin and Tomas Mikolov
Facebook AI Research
{bojanowski, egrave, ajoulin, tmikolov}@fb.com

Abstract

Continuous word representations, trained on large unlabeled corpora are useful for many natural language processing tasks. Popular models that learn such representations ignore the morphology of words, by assigning a distinct vector to each word. This is a limitation, especially for languages with large vocabularies and many rare words. In this paper, we propose a new approach based on the skipgram model, where each word is represented as a bag of character n -grams. A vector representation is associated to each character n -gram; words being represented as the sum of these representations. Our method is fast, allowing to train models on large corpora quickly and allows us to compute word representations for words that did not appear in the training data. We evaluate our word representations on nine different languages, both on word similarity and analogy tasks. By comparing to recently proposed morphological word representations, we show that our vectors achieve state-of-the-art performance on these tasks.

1 Introduction

Learning continuous representations of words has a long history in natural language processing (Rumelhart et al., 1988). These representations are typically derived from large unlabeled corpora using co-occurrence statistics (Deerwester et al., 1990; Schütze, 1992; Lund and Burgess, 1996). A long

et al., 2010; Baroni and Lenci, 2010). In the neural network community, Collobert and Weston (2008) proposed to learn word embeddings using a feed-forward neural network, by predicting a word based on the two words on the left and two words on the right. More recently, Mikolov et al. (2013b) proposed simple log-bilinear models to learn continuous representations of words on very large corpora efficiently.

Most of these techniques represent each word of the vocabulary by a distinct vector, without parameter sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages, such as Turkish or Finnish. For example, in French or Spanish, most verbs have more than forty different inflected forms, while the Finnish language has fifteen cases for nouns. These languages contain many word forms that occur rarely (or not at all) in the training corpus, making it difficult to learn good word representations. Because many word formations follow rules, it is possible to improve vector representations for morphologically rich languages by using character level information.

In this paper, we propose to learn representations for character n -grams, and to represent words as the sum of the n -gram vectors. Our main contribution is to introduce an extension of the continuous skip-



FastText foi baseado no fato de que muitos modelos treinados para gerar tais representações ignoram a morfologia das palavras, gerando um vetor totalmente distinto para cada palavra, mesmo quando elas possuem as mesmas características morfológicas (ex: “quebrar” e “quebrado”).



A representação vetorial é associada a cada n-grama de caracteres e as palavras são representadas pelas somas dessas representações. A formação de palavras costuma seguir regras, o que torna possível gerar representações vetoriais de palavras para linguas ricas morfologicamente usando informação a nível de caractere.



Wang2Vec

- Criado por Wang Ling, Chris Dyer, Alan Black e Isabel Trancoso em 2015;
- Eficiente computacionalmente;
- Algoritmos: CWINDOW e Skip-gram estruturado;
- Frameworks: Gensim

Two/Too Simple Adaptations of Word2Vec for Syntax Problems

Wang Ling Chris Dyer Alan Black Isabel Trancoso

L²F Spoken Systems Lab, INESC-ID, Lisbon, Portugal

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Instituto Superior Técnico, Lisbon, Portugal

{lingwang, cdyer, awb}@cs.cmu.edu

isabel.trancoso@inesc-id.pt

Abstract

We present two simple modifications to the models in the popular Word2Vec tool, in order to generate embeddings more suited to tasks involving syntax. The main issue with the original models is the fact that they are insensitive to word order. While order independence is useful for inducing semantic representations, this leads to suboptimal results when they are used to solve syntax-based problems. We show improvements in part-of-speech tagging and dependency parsing using our proposed models.

1 Introduction

Word representations learned from neural language models have been shown to improve many NLP tasks, such as part-of-speech tagging (Collobert et al., 2011), dependency parsing (Chen and Manning, 2014; Kong et al., 2014) and machine translation (Liu et al., 2014; Kalchbrenner and Blunsom, 2013; Devlin et al., 2014; Sutskever et al., 2014). These low-dimensional representations are learned as parameters in a language model and trained to maximize the likelihood of a large corpus of raw text. They are then incorporated as features along side hand-engineered features (Turian et al., 2010), or used to initialize the parameters of neural networks targeting tasks for which substantially less training data is available (Hinton and Salakhutdinov, 2012; Erhan et al., 2010; Guo et al., 2014).

One of the most widely used tools for building word vectors are the models described in (Mikolov et al., 2013), implemented in the Word2Vec tool,

in particular the “skip-gram” and the “continuous bag-of-words” (CBOW) models. These two models make different independence and conditioning assumptions; however, both models discard word order information in how they account for context. Thus, embeddings built using these models have been shown to capture semantic information between words, and pre-training using these models has been shown to lead to major improvements in many tasks (Collobert et al., 2011). While more sophisticated approaches have been proposed (Dhillon et al., 2011; Huang et al., 2012; Faruqui and Dyer, 2014; Levy and Goldberg, 2014; Yang and Eisenstein, 2015), Word2Vec remains a popular choice due to their efficiency and simplicity.

However, as these models are insensitive to word order, embeddings built using these models are sub-optimal for tasks involving syntax, such as part-of-speech tagging or dependency parsing. This is because syntax defines “what words go where?”, while semantics than “what words go together”. Obviously, in a model where word order is discarded, the many syntactic relations between words cannot be captured properly. For instance, while most words occur with the word *the*, only nouns tend to occur exactly afterwards (e.g. *the cat*). This is supported by empirical evidence that suggests that order-insensitivity does indeed lead to substandard syntactic representations (Andreas and Klein, 2014; Bansal et al., 2014), where systems using pre-trained with Word2Vec models yield slight improvements while the computationally far more expensive which use word order information embeddings of Collobert et al. (2011) yielded much better results.



Wang2Vec

Com o objetivo de possibilitar que o Word2Vec tenha um bom desempenho em tarefas baseadas em sintaxe, Ling et al. (2015) propuseram duas modificações simples no Word2Vec, uma para o modelo Skip-gram e outra para o modelo CBOW.

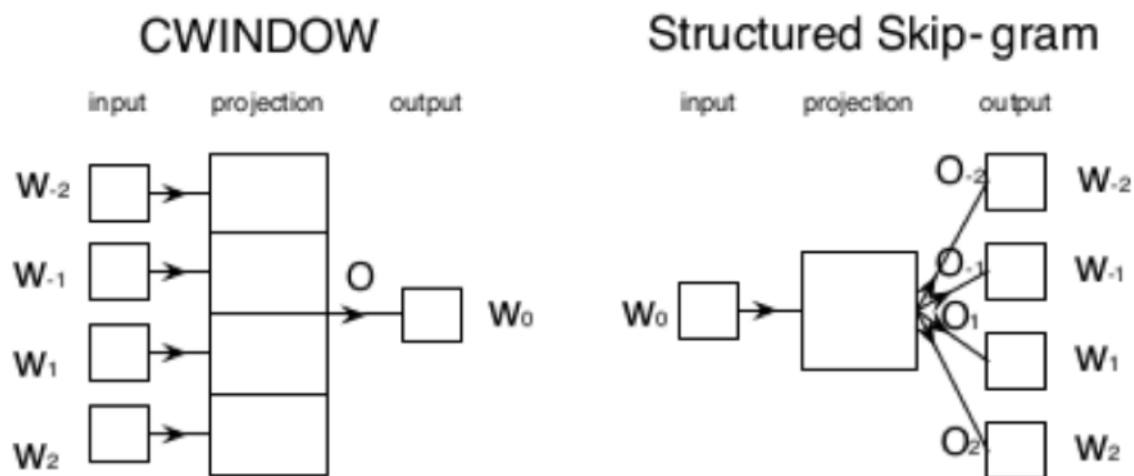
Skip-gram estruturado: define um conjunto de matrizes de saída onde cada uma tem como objetivo prever a saída de uma posição específica de uma palavra de contexto para a palavra alvo.

CWINDOW: são definidas diferentes matrizes de saída que recebem vetores de palavras de contexto concatenados na ordem em que as palavras ocorrem.



Wang2Vec

Melhorar relações sintáticas entre as palavras. Importante para tarefas como etiquetação morfossintática e análise de dependência.



Fonte: Retirado de (LING et al., 2015)



Sense-Specific Word Embeddings

- Um dos papers mais importantes é o de Neelakantan et al. (2015);
- Mais custoso computacionalmente;
- Algoritmos: MSSG e NP-MSSG;
- Frameworks: <https://bitbucket.org/jeevan-shankar/multi-sense-skipgram>

Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space

Arvind Neelakantan*, Jeevan Shankar*, Alexandre Passos, Andrew McCallum
Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA, 01003
{arvind, jshankar, apassos, mccallum}@cs.umass.edu

Abstract

There is rising interest in vector-space word embeddings and their use in NLP, especially given recent methods for their fast estimation at very large scale. Nearly all this work, however, assumes a single vector per word type—ignoring polysemy and thus jeopardizing their usefulness for downstream tasks. We present an extension to the Skip-gram model that efficiently learns multiple embeddings per word type. It differs from recent related work by jointly performing word sense discrimination and embedding learning, by non-parametrically estimating the number of senses per word type, and by its efficiency and scalability. We present new state-of-the-art results in the word similarity in context task and demonstrate its scalability by training with one machine on a corpus of nearly 1 billion tokens in less than 6 hours.

1 Introduction

Representing words by dense, real-valued vector embeddings, also commonly called “distributed representations,” helps address the curse of dimensionality and improve generalization because they can place near each other words having similar semantic and syntactic roles. This has been shown dramatically in state-of-the-art results on language modeling (Bengio et al, 2003; Mnih and Hinton, 2007) as well as improvements in other

of Mikolov et al (2013a); Mikolov et al (2013b)—relatively simple log-linear models that can be trained to produce high-quality word embeddings on the entirety of English Wikipedia text in less than half a day on one machine.

There is rising enthusiasm for applying these models to improve accuracy in natural language processing, much like Brown clusters (Brown et al, 1992) have become common input features for many tasks, such as named entity extraction (Miller et al, 2004; Ratnikov and Roth, 2009) and parsing (Koo et al, 2008; Täckström et al, 2012). In comparison to Brown clusters, the vector embeddings have the advantages of substantially better scalability in their training, and intriguing potential for their continuous and multi-dimensional interrelations. In fact, Passos et al (2014) present new state-of-the-art results in CoNLL 2003 named entity extraction by directly inputting continuous vector embeddings obtained by a version of Skip-gram that injects supervision with lexicons. Similarly Bansal et al (2014) show results in dependency parsing using Skip-gram embeddings. They have also recently been applied to machine translation (Zou et al, 2013; Mikolov et al, 2013c).

A notable deficiency in this prior work is that each word type (*e.g.* the word string plant) has only one vector representation—polysemy and homonymy are ignored. This results in the word plant having an embedding that is approximately the average of its different contextual semantics relating to biology, placement, manufacturing and power generation. In moderately high-dimensional spaces a vector can be relatively “close” to multiple regions at a time, but this does



Sense-Specific Word Embeddings

Apesar de muito úteis em diversas aplicações, os vetores de palavras citados anteriormente têm limitações!

A geração de vetores de palavras específicos de sentido é uma linha de pesquisa recente, que surgiu a partir da constatação da limitação dos métodos tradicionais, como o Word2Vec de Mikolov et al. (2013), que geram apenas um vetor por palavra, desconsiderando que uma mesma palavra pode possuir sentidos (significados) diferentes.

MSSG (*Multiple-sense Skip-gram*): implementa uma quantidade fixa de sentidos possíveis para cada palavra

NP-MSSG (*Non-parametric Multiple-sense Skip-gram*): faz a descoberta de quantidade de sentidos por palavra em tempo de execução.



Sense-Specific Word Embeddings

- Baseados em conhecimento
- Não supervisionados.

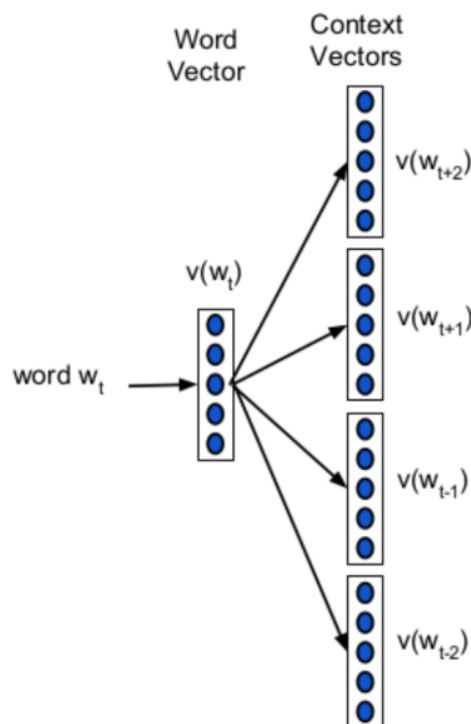
Exemplo:

- A **letra** (sentido1) da canção é excêntrica.
- A música tem uma ótima **letra** (sentido1).
- Sua **letra** (sentido2) é muito bonita.
- A minha **letra** (sentido2) cursiva é uma obra de arte.

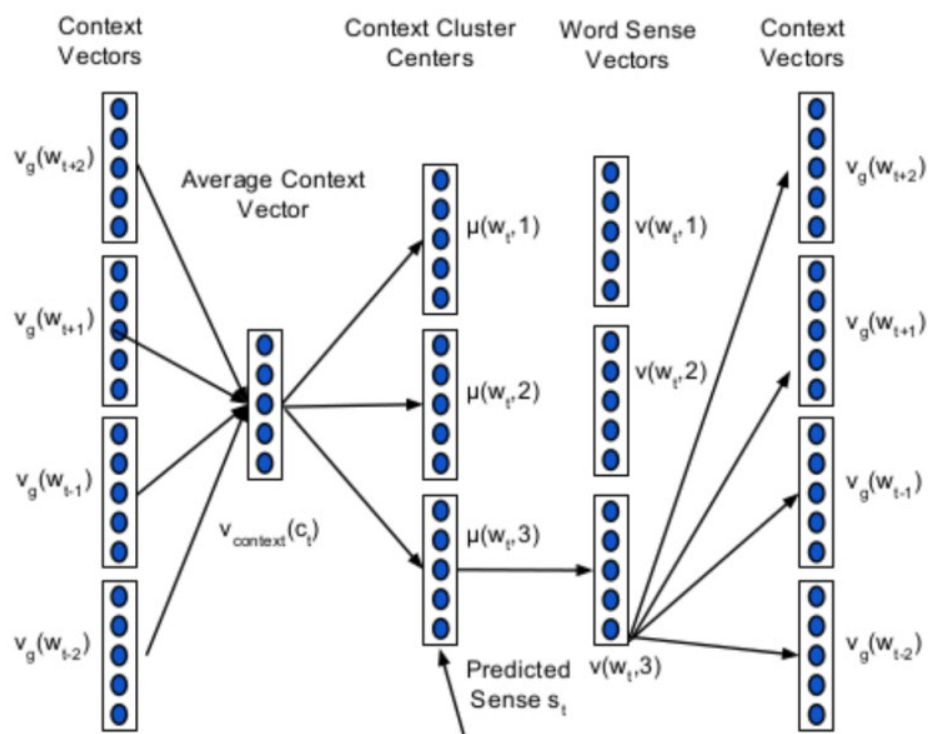


Sense-Specific Word Embeddings

Na primeira imagem tem-se a estrutura do Skip-gram original. Na segunda tem-se a arquitetura do MSSG.



Fonte: Retirado de (NEELAKANTAN et al., 2015)



Fonte: Retirado de (NEELAKANTAN et al., 2015)



Referências

BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. Journal of machine learning research, v. 3, n. Feb, p.1137–1155, 2003.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.

HARRIS, Z. Distributional structure. Word, v. 23, n. 10, p. 146–162, 1954.

JURAFSKY, D.; MARTIN, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. [S.l.]: MIT Press, 2008.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: International Conference on Machine Learning. [S.l.: s.n.], 2014. p. 1188–1196.

LING, W.; DYER, C.; BLACK, A. W.; TRANCOSO, I. Two/too simple adaptations of word2vec for syntax problems. In: HLT-NAACL. [S.l.: s.n.], 2015. p. 1299–1304.



Referências

MIKOLOV, T.; KOPECKY, J.; BURGET, L.; GLEMBEK, O. et al. Neural network based language models for highly inflective languages. In: IEEE. Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on. [S.l.], 2009. p. 4725–4728.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

NEELAKANTAN, A.; SHANKAR, J.; PASSOS, A.; MCCALLUM, A. Efficient non-parametric estimation of multiple embeddings per word in vector space. arXiv preprint arXiv:1504.06654, 2015.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). [S.l.: s.n.], 2014. p. 1532–1543.



Links

Vídeo Geoffrey Hinton:

https://www.youtube.com/watch?v=fDR1I2Shw_E

Vídeo Yann LeCun:

<http://techtalks.tv/talks/whats-wrong-with-deep-learning/61639/>

Repositório de Embeddings (Artigo):

<https://arxiv.org/abs/1708.06025>

Repositório de Embeddings:

<http://nilc.icmc.usp.br/embeddings>

OBRIGADA!

PATROCÍNIO:



INFOMACH
TECNOLOGIA PARA NEGÓCIOS

Saúdemobi



www.deeplearning.com.br

APOIO:



DATA^H
01