详解深度语义匹配模型DSSM

小小挖掘机 2020-01-15

以下文章来源于有三AI ，作者小Dream哥

有三AI

三人行必有AI，本公众号聚焦于让大家能够系统性地完成AI各个领域所需的专业知识的...

> 所谓语义匹配，就是在语义上衡量文本的相似度，在产业界有很多的应用需求。例如，在FAQ场景中需要计算用户输入与标问之间的相似度来寻找合适的答案。本文介绍一种经典的语义匹配技术，DSSM，主要用于语料的召回和粗排。
>
> 作者&编辑 | 小Dream哥

1 DSSM的提出

较早期的语义匹配模型都是基于关键词的匹配，例如LSA等，无法匹配语义层面的信息。基于此，DSSM（Deep Structured Semantic Models）提出深度语义匹配模型，期望能够在语义层面匹配query之间的相似性。

顾名思义，DSSM是一种用于语义相似度计算的深度网络，我们来看看它的庐山真面目到底是怎么样的。

2 整体看结构

我们先来整体来看一下DSSM的网络结构，以整体上对它有一个把握和感觉。如下图所示，是DSSM的网络架构图：

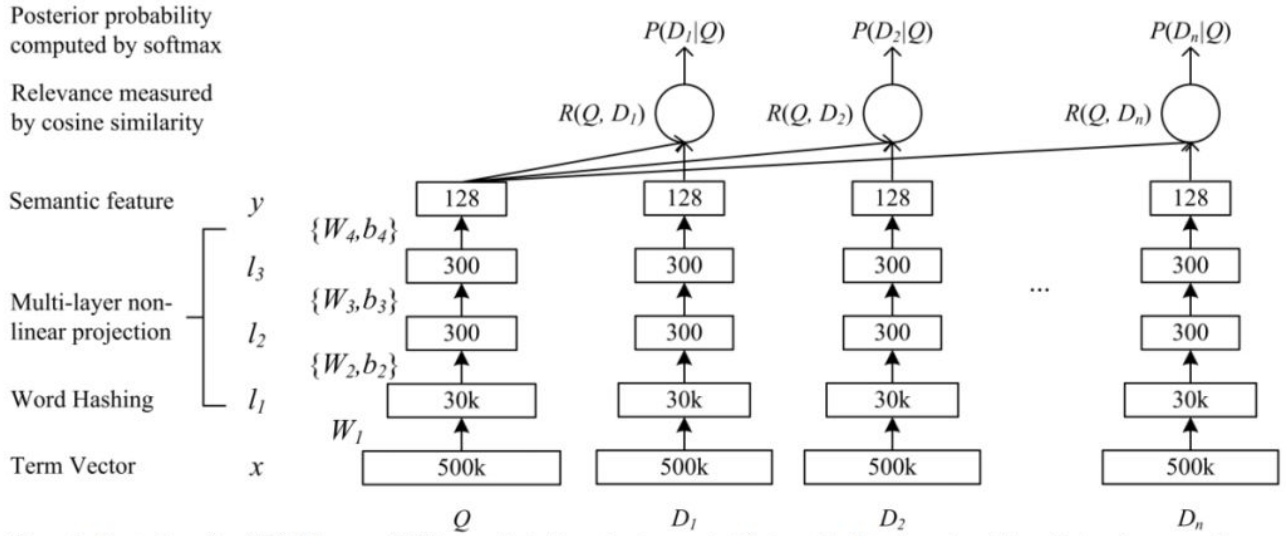


Figure 1: Illustration of the DSSM. It uses a DNN to map high-dimensional sparse text features into low-dimensional dense features in a semantic space. The first hidden layer, with 30k units, accomplishes word hashing. The word-hashed features are then projected through multiple layers of non-linear projections. The final layer's neural activities in this DNN form the feature in the semantic space.

论文原文：https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/cikm2013_DSSM_fullversion.pdf

整体上来看，DSSM网络总共有6层：

1.第一层是输入层，DSSM用的词袋模型，后面再详细介绍；
2.第二层经过word hashing，将维度由500K降为30K；
3.第三，四，五层是3个全连接层，通过这三个全连接层，进行语义特征的提取，并降维度降低到128维；
4.第六层为输出层，计算Q和D之间的余弦相似度之后，输出他们之间的相似度。

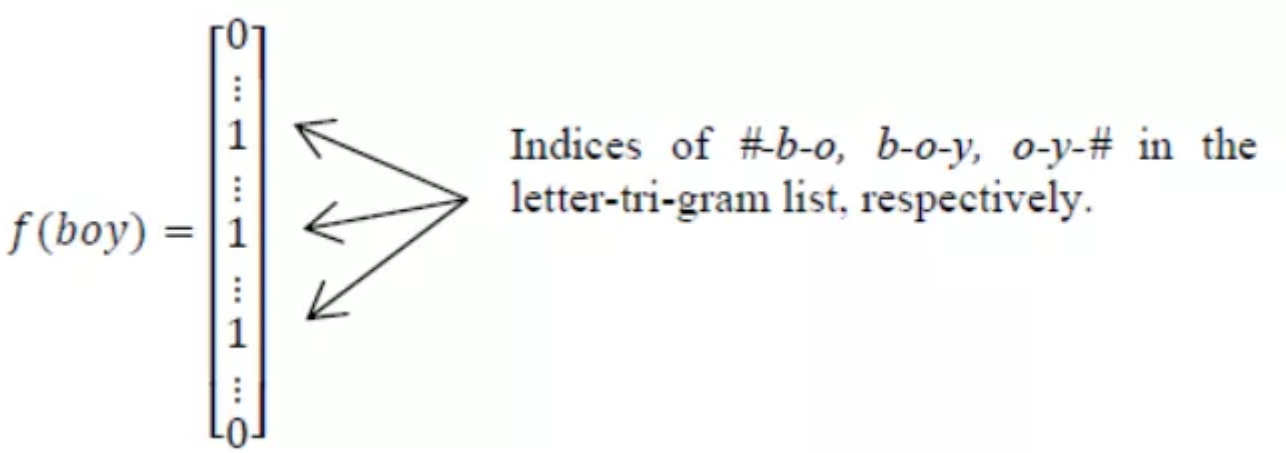
3 输入层及word hashing

DSSM的输入层结合了词哈希（word hashing）和语义匹配，我们在讲词向量的时候详细介绍了词袋模型，忘记的同学可以点击如下链接先了解：

【NLP-词向量】词向量的由来及本质

总的来说词袋模型就是把文本看成是一个装着词的袋子，记录一个文本中，有这个词几个，那个词几个。前提到过，当词典非常大时，用词袋模型会造成维度灾难。所以DSSM还引入了word hashing。

Word hashing主要目的是为了减少维度，在英文里，采用letter-ngams来对单词进行切分，如下图所示，加入采用letter-trigams来对词进行切分，则boy这个词可以切分为（#bo,boy,oy#）三个。按这个方法，再将上述词袋里的进行转化。因为英文只有26个字母，这样可以极大的减少维度，如论文中所示将维度从500K转化为30K。



也许反应快的同学很快就会问，英文可以这样做，但是好像中文没有办法这样处理呀？总不能按照偏旁来拆吧？当然不会按照偏旁来拆了，加入汉字部首偏旁特征的研究目前还不很成功。

那么中文怎么处理呢？其实很简单，在单纯的DSSM模型中，中文是按照“**字袋模型**”来处理的，参考词袋模型，也就是将文本转化成，有几个某某字，有几个某某字。因为中文字个数是有限的，常用的字大概有15K左右，因此这种做法不会有维度过大的问题。

4 特征提取层和相似度计算

熟悉深度学习的朋友，应该很容易看明白DSSM的特征抽取层，其实就是3个全连接层串行的连接起来。看看数学：

$$l_i = f(W_i l_{i-1} + b_i), i = 2, \dots, N-1 \quad (3)$$

$$y = f(W_N l_{N-1} + b_N)$$

where we use the *tanh* as the activation function at the output layer and the hidden layers $l_i, i = 2, \dots, N-1$:

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (4)$$

可以看出，在DSSM中采用tanh作为激活函数。

通过计算各个Q及D的特征表征，得到了一些128维的特征向量。随后在DSSM中，通过计算Q和D之间的余弦距离来评价他们之间相似度，计算公式如下图所示：

$$R(Q,D) = \text{cosine}(y_Q, y_D) = \frac{y_Q{}^T y_D}{\|y_Q\| \|y_D\|}$$

5 DSSM的训练

那么DSSM训练的过程是怎么样的呢？细心的同学会发现，DSSM网络结构图中，DSSM的输入是一个Querry和一个文本集**D**，**D**中包含正样本和负样本。

$$P(D|Q) = \frac{\exp(\gamma R(Q,D))}{\sum_{D' \in \boldsymbol{D}} \exp(\gamma R(Q,D'))}$$

其中 r 为 softmax 的平滑因子，D 为 Query 下的正样本，**D** 为 Query 下的整个样本空间。

上述公式，计算一个样本空间内正样本的平滑概率，R(Q,D)为两个文本之间余弦距离。

在训练阶段，通过极大似然估计，最小化损失函数为：

$$L(\Lambda) = -\log \prod_{(Q,D^+)} P(D^+|Q)$$