单词相似度计算与短文本对话的研究

林连升

December 21, 2017

主要工作

- **Word Similarity at NLPCC-2016**
- **DBQA at NLPCC-2017**
  Document-Based Question Answering
- **STC-2 at NTCIR-13**
  Short Text Conversation

Word Similarity

数据集：PKU-500

Table 1. The top 10 similar word pairs

word 1	word 2	score
WTO	世界贸易组织	10
紫禁城	故宫	10
计算机	电脑	9.9
赢	胜	9.8
维他命	维生素	9.5
化肥	化学肥料	9.5
课程表	课表	9.5
互联网	因特网	9.5
假货	赝品	9.5

Table 2. The top 10 dissimilar word pairs

word 1	word 2	score
讲价	打架	1
教授	黄瓜	1
控制	恐高	1
阻力	天花板	1
结盟	无理取闹	1.1
调查	努力	1.1
玻璃	魔术师	1.1
干扰	上网	1.1
课堂	美食	1.1

数据集：PKU-500

Table 3. The top 10 word pairs with low standard deviation

word 1	word 2	score	Stad.
教授	黄瓜	1	0.00
控制	恐高	1	0.00
阻力	天花板	1	0.00
WTO	世界贸易组织	10	0.00
紫禁城	故宫	10	0.00
讲价	打架	1	0.22
玻璃	魔术师	1.1	0.30
干扰	上网	1.1	0.30
课堂	美食	1.1	0.30

Table 4. The top 10 word pairs with high standard deviation

word 1	word 2	score	Stad
没戏	没辙	4.9	3.03
只管	尽管	4	2.94
GDP	生产力	6.5	2.80
包袱	段子	2.6	2.71
日期	时间	6	2.67
由此	通过	3.4	2.66
爱面子	好 高 骛远	4	2.56
一方面	一边	5.4	2.54
托福	GRE	8	2.54

评测指标

Spearman's rank correlation coefficient:

$$p = 1 - \frac{6 \sum_{i=1}^{n} (R_{Xi} - R_{Yi})^2}{n(n^2 - 1)} \tag{1}$$

where n is the number of word pairs being evaluated, R_{Xi} and R_{Yi} are the standard deviations of the rank of automatic computing results and human labelled scores, respectively.

NLPCC-2016 评测结果

Table 5. The overall evaluation results

Team ID	Organization	Spear.
SXUCFN-QA	Shanxi University	0.518
DLUT_NLPer	Dalian University of Technology	0.457
CQUT_AC996	Chongqing University of Technology	0.436
nlp_polyu	The Hong Kong Polytechnic University	0.421
BLCU_CNLR	Beijing Language and Culture University	0.414
Cbrain	Institute of Automation, Chinese Academy of Sciences	0.412
CIST	Beijing University of Posts and Telecommunications	0.405
wanghao.ftd	Shanghai Jiao Tong University	0.405
DUTNLP	Dalian University of Technology	0.372
BIT_CWSM	Beijing Institute of Technology	0.371
NJUST-CWS	Nanjing University of Science and Technology	0.365
TJIIP	Tongji University	0.357
SWJTU_CCIT	Southwest Jiaotong University	0.349

我的结果

Method	Spearman
NLPCC-2016 best result	0.518
Word2Vec1(2G 新闻语料，12 对词未覆盖)	0.338
Word2Vec2(2G 新闻语料 + 补充语料)	0.400
Word2Vec3(20G 语料)	0.426
知网 1(刘群等人，2002，118 对词未覆盖)	0.258
知网 2(刘群等人，2002，未覆盖给 0.5 的值)	0.418
同义词词林（田久乐等人，2010）	0.406
同义词词林（朱新华等人，2016）	0.431
Word2Vec2 + 同义词词林 2010	**0.518**
Word2Vec2 + 同义词词林 2016	**0.538**
Word2Vec2 + 同义词词林 2010 + 知网 1	**0.582**
Word2Vec2 + 同义词词林 2016 + 知网 1	**0.587**

同义词词林

参考两篇 Paper:

1. 田久乐, 赵蔚. 基于同义词词林的词语相似度计算方法 [J]. 吉林大学学报 (信息科学版),2010,06:602-608.
2. 朱新华, 马润聪, 孙柳, 陈宏朝. 基于知网与词林的词语语义相似度计算 [J]. 中文信息学报,2016,04:29-36.

同义词词林简介

Ba01A02= 物质 质 素

Cb02A01= 东南西北 四方

Ba01A03@ 万物

Cb06E09@ 民间

Ba01B08# 固体 液体 气体 流体 半流体

Ba01B10# 导体 半导体 超导体

表 2 词语编码表

编码位	1	2	3	4	5	6	7	8
符号举例	D	a	1	5	B	0	2	= \ # \ @
符号性质	大类	中类	小类		词群	原子词群		
级别	第 1 级	第 2 级	第 3 级		第 4 级	第 5 级		

田久乐等的算法，2010

- 主要思想：基于同义词词林的结构，利用词语中义项的编号，根据两个义项的语义距离，计算出义项相似度。
- 首先判断在同义词林中作为叶子节点的两个义项在哪一层开始分支。例如：Aa01A01 与 Aa01B01，即在第 4 层分支。在每一层分支，分别对应一个系数，然后再乘以调节参数。

田久乐等的算法，2010

若两个义项的相似度用 Sim 表示

1）若两个义项不在同一棵树上

$$\mathrm{Sim}(A, B) = f; \tag{1}$$

2）若两个义项在同一棵树上:

¹ 若在第 2层分支，系数为 a

$$\mathrm{Sim}(A, B) = 1 \times a \times \cos\left(n \times \frac{\pi}{180}\right)\left(\frac{n-k+1}{n}\right) \tag{2}$$

④若在第 3层分支，系数为 b

$$\mathrm{Sim}(A, B) = 1 \times 1 \times b \times \cos\left(n \times \frac{\pi}{180}\right)\left(\frac{n-k+1}{n}\right) \tag{3}$$

㈣在第 4层分支，系数为 c

$$\mathrm{Sim}(A, B) = 1 \times 1 \times 1 \times c \times \cos\left(n \times \frac{\pi}{180}\right)\left(\frac{n-k+1}{n}\right) \tag{4}$$

¼ 在第 5层分支，系数为 d

$$\mathrm{Sim}(A, B) = 1 \times 1 \times 1 \times 1 \times d \times \cos\left(n \times \frac{\pi}{180}\right)\left(\frac{n-k+1}{n}\right) \tag{5}$$

Figure 1: 这里 n 是分支数，k 是两个分支的距离，abcdf 是实验得到的系数。

田久乐等的算法，2010

举个例子:

Ae05A01= 邮递员 邮差 信差 信使 绿衣使者 通信员 投递员

Ae05A02= 交通员 交通 通讯员

Ae05A03= 联络员 联络官 联系人

其中，义项“邮递员”和“联络员”在第五层分支，分支数 n=3，分支距离 k=2，其相似度为:

sim = d * cos(n*pi/180) * ((n-k+1)/n)

= 0.96 * cos(3*pi/180)*((3-2+1)/3)

= 0.639

田久乐等的算法，2010

特殊情况:
如果五层编码都相同，则靠末尾的 = # @ 来计算。

- = 表示同义，相似度取 1
- # 表示相关，相似度取 e(e = 0.5)
- @ 表示封闭，只有一个词。因此不存在两个词编号相同且末尾为 @ 的情况。

Ab02A08@ 老太公
Ab02B01= 成年人 壮年人 大人 人丁 壮丁 佬 中年人
Ab02C01= 老小 老少 大小 老幼 老老少少 白叟黄童 大大小小
Ab02C02# 遗老 遗少 遗老遗少 封建残余

朱新华等的算法，2016

- （田久乐等，2010）的算法是以分支节点数 n 和分支间隔 k 为主要考虑因素，因此会出现许多距离近的词语因分支间隔远而算出相似度过低的不合理现象。为解决这一问题，朱新华等提出了一个以词语距离 d 为主要影响因素、分支节点数 n 和分支间隔 k 为调节参数的计算公式。

$$sim(C_1, C_2) = (1.05 - 0.05 dis(C_1, C_2)) \sqrt{e^{\frac{-k}{2n}}}$$

朱新华等的算法，2016

- 其中，$dis(C_1, C_2)$ 是词语编码 C_1, C_2 在树状结构中的距离函数。W_1、W_2、W_3、W_4 分别为 0.5、1、2.5、2.5。因此，距离有 1、3、8、13 几种情况。

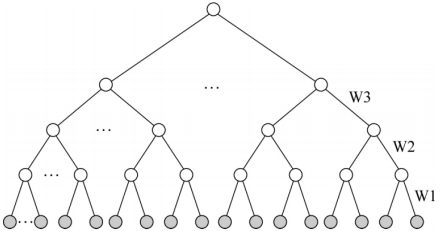

图 2　同义词词林的 5 层树形结构

《知网》简介

- 《知网》的结构
  - “概念”，是用一种“知识描述语言”来描述的，可以理解为一个词的一种意思。
  - “义原”，是用于描述一个“概念”的最小语义单位。

表 2：《知网》知识描述语言实例

词	概念编号	描述语言
打	017144	exercise|锻练,sport|体育
男人	059349	human|人,family|家,male|男
高兴	029542	aValue|属性值,circumstances|境况,happy|福,desired|良
生日	072280	time|时间,day|日,@ComeToWorld|问世,$congratulate|祝贺
写信	089834	write|写,ContentProduct=letter|信件
北京	003815	place|地方,capital|国都,ProperName|专,(China|中国)
爱好者	000363	human|人,*FondOf|喜欢,#WhileAway|消闲
必须	004932	{modality|语气}
串	015204	NounUnit|名量,&(grape|葡萄),&(key|钥匙)

[3] 刘群, 李素建. 基于《知网》的词汇语义相似度计算 [J]. 中文计算语言学, 2002.

知识描述语言的符号及其含义

表 1: 《知网》知识描述语言中的符号及其含义

,	多个属性之间，表示“和”的关系
#	表示“与其相关”
%	表示“是其部分”
$	表示“可以被该‘V’处置，或是该“V”的受事，对象，领有物，或者内容
*	表示“会‘V’或主要用于‘V’，即施事或工具
+	对 V 类，它表示它所标记的角色是一种隐性的，几乎在实际语言中不会出现
&	表示指向
~	表示多半是，多半有，很可能的
@	表示可以做“V”的空间或时间

义原的分类

- 《知网》一共采用了个 1500 义原，这些义原分为以下几个大类:
  1. Event| 事件
  2. entity| 实体
  3. attribute| 属性
  4. aValue| 属性值
  5. quantity| 数量
  6. qValue| 数量值
  7. SecondaryFeature| 次要特征
  8. syntax| 语法
  9. EventRole| 事件角色
  10. EventFeatures| 事件属性
- 这 10 类义原又分为三组:
  - 1-7 基本义原
  - 8 语法义原
  - 9、10 关系义原

义原的层次结构

- 在《知网》中，一共描述了义原之间的 8 种关系：上下位关系、同义关系、反义关系、对义关系、属性-宿主关系、部件-整体关系、材料-成品关系、事件-角色关系。
- 根据义原中最重要的上下位关系，所有的“基本义原”组成了一个义原层次体系。这个义原层次体系是一个树状结构，也是《知网》进行语义相似度计算的基础。

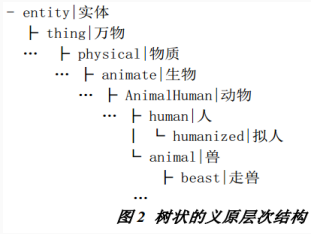

图 2 树状的义原层次结构

知识描述语言归纳

- 《知网》收录的词语主要分为两类，一类是实词，一类是虚词
- 虚词的描述比较简单，用“句法义原”或“关系义原”进行描述
- 实词的描述比较复杂，由一系列用逗号隔开的“语义描述式”组成，这些“语义描述式”又有以下三种形式：
  - 基本义原描述式：用“基本义原”进行描述；
  - 关系义原描述式：用“关系义原 = 基本义原”或者“关系义原 =(具体词)”或者“(关系义原 = 具体词)”来描述；
  - 关系符号描述式：用“关系符号 基本义原”或者“关系符号 (具体词)”加以描述。

基于《知网》的词语语义相似度计算方法

- 词语相似度计算
- 义原相似度计算
- 虚词概念的相似度计算
- 实词概念的相似度计算

词语相似度计算

对于两个汉语词语 W_1 和 W_2，如果 W_1 有 n 个概念：S_1^1，S_2^1，…，S_n^1，W_2 有 m 个概念：S_1^2，S_2^2，…，S_m^2，规定它们的相似度是各个概念的相似度之最大值：

$$Sim(W_1, W_2) = \max_{i=1...n, j=1...m} Sim(S_i^1, S_j^2) \tag{2}$$

义原相似度计算

- 由于所有的概念都最终归结于用义原来表示，所以义原的相似度计算是概念相似度计算的基础。
- 假设两个义原在树状的层次体系中的路径距离为 d，我们可以得到这两个义原之间的相似度为：

$$Sim(p_1, p_2) = \frac{\alpha}{d + \alpha} \tag{3}$$

虚词概念的相似度计算

- 我们认为，在实际的文本中，虚词和实词总是不能互相替换的，因此，虚词概念和实词概念的相似度总是为零。
- 由于虚词概念总是用“语法义原”或“关系义原”这两种方式进行描述，所以，虚词概念的相似度计算归结为，计算其对应的“句法义原”或“关系义原”之间的相似度即可。

实词概念的相似度计算

实词概念的描述包括两种抽象的结构：集合、特征结构

- 集合
  打：{exercise| 锻炼，sport| 体育}
- 特征结构
  写信：{write| 写，ContentProduct=letter| 信件}

集合之间的相似度计算

1. 首先计算两个集合的所有元素两两之间的相似度；
2. 从所有的相似度值中选择最大的一个记下，将这个相似度值对应的两个元素删除；
3. 重复上述第 2 步，直到所有的元素都被删除；
4. 没有建立起对应关系的元素与空元素对应。

集合之间的相似度等于其元素对的相似度的平均值

特征结构之间的相似度计算

特征结构是“属性-值”对的集合

- 将两个特征结构之间相同的属性一一对应
- 计算每个“属性-值”对的相似度，即计算其“值”的相似度
- 特征结构之间的相似度就等于所有“属性-值”对的相似度的平均值

实词概念的相似度计算

实词概念的描述可以表示为一个特征结构，包括以下四个特征：

- 第一基本义原：其值为一个基本义原，这一部分的相似度记为 $Sim_1(S_1, S_2)$；
- 其它基本义原：除第一基本义原以外的所有基本义原，其值为一个基本义原的集合，这一部分的相似度记为 $Sim_2(S_1, S_2)$；
- 关系义原描述：对应所有的关系义原描述式，其值是一个特征结构，对于该特征结构的每一个特征，其属性是一个关系义原，其值是一个基本义原，或一个具体词。这一部分的相似度记为 $Sim_3(S_1, S_2)$；
- 关系符号描述：对应所有的关系符号描述式，其值也是一个特征结构，对于该特征结构的每一个特征，其属性是一个关系义原，其值是一个集合，该集合的每个元素是一个基本义原，或一个具体词。这一部分的相似度记为 $Sim_4(S_1, S_2)$。

实词概念的相似度计算

- 两个概念的总体相似度为：

$$Sim(S_1, S_2) = \sum_{i=1}^{4} \beta_i Sim_i(S_1, S_2) \tag{4}$$

- 其中，$\beta_i (1 \leq i \leq 4)$ 是可调节的参数，且有：$\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$，$\beta_1 \geq \beta_2 \geq \beta_3 \geq \beta_4$
- 后者反映了 Sim_1 到 Sim_4 对于总体相似度所起到的作用依次递减。

Open Domain Question Answering

- **NLPCC**
  Natural Language Processing and Chinese Computing

- **Open Domain Question Answering**
  including three tasks:
  - Knowledge-Based Question Answering (KBQA)
  - **Document-Based Qestion Answering (DBQA)**
  - Table-Based Question Answering (TBQA)

The task of DBQA is to answer Chinese questions by selecting one or multiple sentences from a given document as answers.

俄罗斯贝加尔湖的面积有多大？ \t 贝加尔湖，中国古代称为北海，位于俄罗斯西伯利亚的南部。 \t **0**

俄罗斯贝加尔湖的面积有多大？ \t 贝加尔湖是世界上最深，容量最大的淡水湖。 \t **0**

俄罗斯贝加尔湖的面积有多大？ \t 贝加尔湖贝加尔湖是世界上最深和蓄水量最大的淡水湖。 \t **0**

俄罗斯贝加尔湖的面积有多大？ \t 它位于布里亚特共和国 (Buryatiya) 和伊尔库茨克州 (Irkutsk) 境内。 \t **0**

俄罗斯贝加尔湖的面积有多大？ \t 湖型狭长弯曲，宛如一弯新月，所以又有“月亮湖”之称。 \t **0**

俄罗斯贝加尔湖的面积有多大？ \t 贝加尔湖长 636 公里，平均宽 48 公里，最宽 79.4 公里，面积 3.15 万平方公里。 \t **1**

俄罗斯贝加尔湖的面积有多大？ \t 贝加尔湖湖水澄澈清冽，且稳定透明（透明度达 40.8 米），为世界第二。 \t **0**

Figure 2: An example for DBQA

Distance-based Word Overlap

1. Find question word in question and get its position p
   俄罗斯 贝加尔湖 的 面积 有 多大
2. Remove stopwords. Get question word list W.
   俄罗斯 贝加尔湖 面积 多大
   0 1 2 3
3. Assign score for each word W

$$s_i = \begin{cases} 2^{-|i-p|} & i \neq p \\ 0 & i = p \end{cases} \tag{5}$$

4. Scoring sentence ans

$score(ans) = \sum_i^l \delta_i, \quad \delta_i = \begin{cases} s_i & w_i \in ans \\ 0 & w_i \notin ans \end{cases}$

l is the length of W, w_i refers to the word in W.

According to whether the word w_i is at the right of the key word or at the left, we update the score computation for the word in W as Eq.4

$$s_i = \begin{cases} \beta \times 2^{-|i-d|} & i > d \\ 2^{-|i-d|} & i < d \\ 0 & i = d \end{cases} \tag{6}$$

β is a positive number tuned by training data. In our experiment, $\beta = 4.3$ is optimized.

Table 1: The result on training data

Method	MAP	MRR
Machine Translation	0.2410	0.2412
Average Word Embedding	0.4610	0.4610
Paraphrase	0.4886	0.4906
Word Overlap	0.5114	0.5134
Distance-based Overlap	0.6848	0.6874
Weighted Distance-based Overlap	0.7266	0.7293

Final Evaluation Results: Document-based QA Task

Rank	Team	TeamID	MRR	MAP	ACC@1
1	复旦大学	KEngine	0.720194	0.716594	0.592
2	北京邮电大学	Prisers	0.689619	0.68576	0.5556
3	同济大学	CU-KG	0.685011	0.680963	0.5512
4	DeepIntell	run2	0.683674	0.680067	0.5492
5	天津大学	TJUNLP	0.677203	0.673271	0.54
6	DeepIntell	run3	0.675772	0.670828	0.5356
7	DeepIntell	primary	0.672872	0.668659	0.5372
8	DeepIntell	run4	0.664586	0.660256	0.5244
9	华中师范大学	CCNU-NLP-Blaze	0.660674	0.658893	0.5144
10	北京大学	ICL-WLL	0.652062	0.649218	0.5056
11	浙江大学	CS-NLP	0.583311	0.580741	0.4284
12	网龙网络有限公司	Nders	0.557158	0.556341	0.3996
13	北京航空航天大学	BHU	0.54846	0.545021	0.372
14	国防科技大学	FuRongWang	0.533575	0.531831	0.3692
15	华东师范大学	ECNU	0.506718	0.503114	0.3404
16	北京联合大学	TensorRQ	0.494292	0.491736	0.3288
17	北京大学	name_system	0.436557	0.434162	0.2696
18	大连理工大学-1	DLUT_NLPer	0.402115	0.40085	0.2172
19	大连理工大学-2	DLUT_NLPer	0.384112	0.382343	0.2016
20	大连理工大学-3	DLUT_NLPer	0.384112	0.382343	0.2016
21	重庆大学	CQUT_AC996	0.353259	0.352269	0.1744

Short Text Conversation

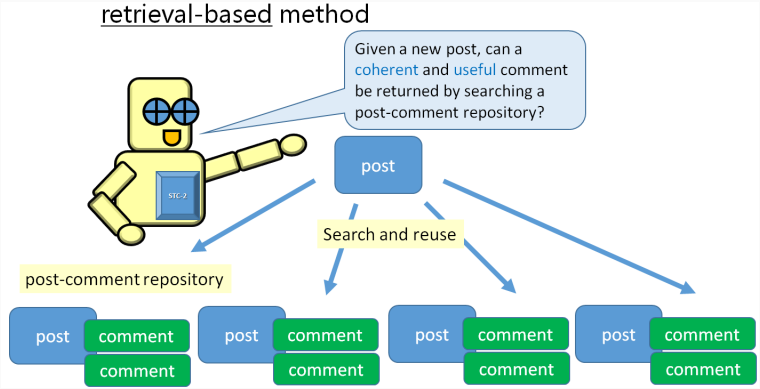

Figure 3: 基于检索的方法

Relevance Assessments

Judging criteria:

1. Fluent
2. Coherent: logically and topically relevant
3. Self-sufficient
4. Substantial

If (1) or (2) is false: label = "L0"
else if (3) or (4) is false: label = "L1"
else: label = "L2"

Labels

Post	意大利禁区里老是八个人...太夸张了吧 There are always 8 Italian players in their own restricted area…Unbelievable!	Related Criteria	Labels
Comment1	我是意大利队的球迷，等待比赛开始。 I am a big fan of the Italy team, waiting for the football match to start	(2) Coherent	L0
Comment2	意大利的食物太美味了 Italian food is absolutely delicious.	(2) Coherent	L0
Comment3	太夸张了吧！ Unbelievable!	(4) Substantial	L1
Comment4	哈哈哈仍然是0：0。还没看到进球。 Haha, it is still 0:0, no goal so far.	(3) Self-sufficient	L1
Comment5	这正是意大利式防守足球。 This is exactly the Italian defending style football game	———	L2

Figure 4: Example of a post and its five candidate comments with human annotation.

System Architecture

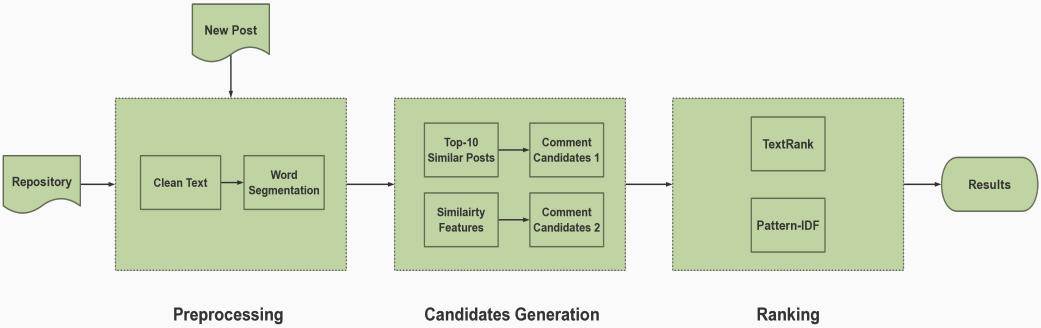

Figure 5: System Architecture

Preprocessing

- Traditional-Simplified Chinese conversion
- Convert Full-width characters into half-width ones
- Word segmentation (PKU standard)
- Replace number, time, url with token <_NUM>, <_TIME>, <_URL> respectively
- Filter meaningless words and special symbols

Short Text ID	test-post-10440
Raw Text	去到美國，还是吃中餐！宫保雞丁家的感覺～ Go to the USA, still eat Chinese food, Kung Pao Chicken, feeling like at home
Without T-S Conversion	去 到 美 國 , 还 是 吃 中餐 ! 宫 保 雞 丁 家 的 感 覺 ~
With T-S Conversion	去 到 美国 , 还 是 吃 中餐 ! 宫保鸡丁 家 的 感觉 ~
Clean Result	去 到 美国 还 是 吃 中餐 宫保鸡丁 家 的 感觉

Short Text ID	test-post-10640
Raw Text	汶川大地震9周年：29个让人泪流满面的瞬间。 9th Anniversary of Wenchuan Earthquake: 29 moments making people tearful
Without token replacement	汶川 大 地震 9 周年 : 29 个 让 人 泪流满面 的 瞬间 。
With token replacement	汶川 大 地震 <_NUM> 周年 : <_NUM> 个 让 人 泪流满面 的 瞬间 。
Clean Result	汶川 大 地震 <_NUM> 周年 <_NUM> 个 让 人 泪流满面 的 瞬间

Similarity Features

- TF-IDF
- LSA (Latent Semantic Analysis)
- LDA (Latent Dirichlet Allocation)
- Word2Vec (skip-gram)
- **LSTM-Sen2Vec**

We combine each post with its corresponding comments to be a document, then train LSA and LDA models on these documents.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (7)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (8)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (9)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (10)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * tanh(C_t) \quad (12)$$

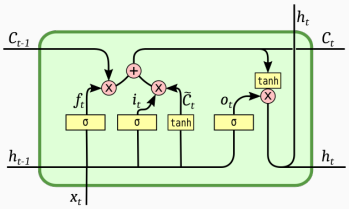

Figure 6: The LSTM Cell.

Mikolov, Toma's. Statistical Language Models Based on Neural Networks. Ph.D. thesis, Brno University of Technology.(2012)

Zaremba, Wojciech, I. Sutskever, and O. Vinyals. Recurrent Neural Network Regularization. Eprint Arxiv (2014).

Attention weight

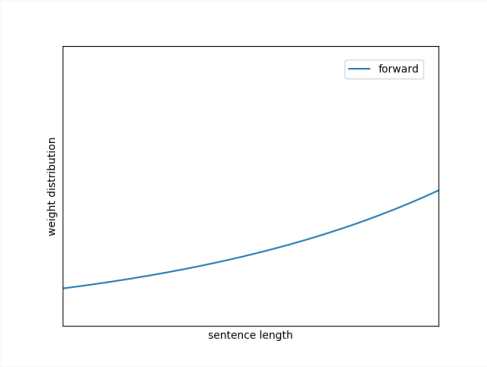

Figure 7: Unidirectional weight distribution

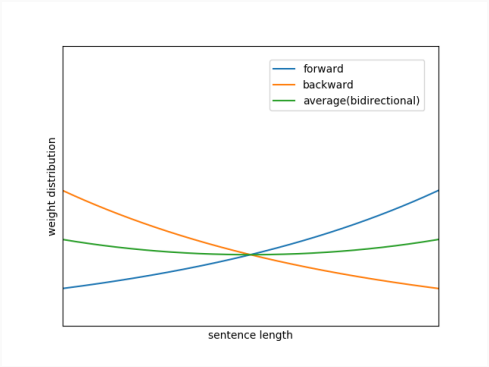

Figure 8: bidirectional weight distribution

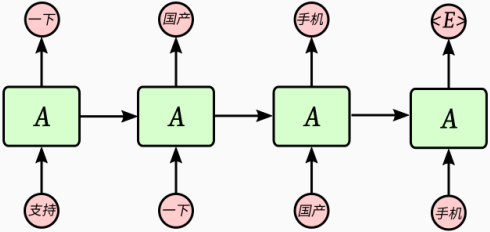

Figure 9: The Unidirectional LSTM

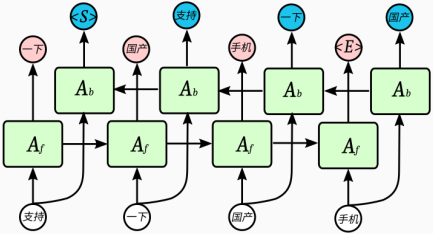

Figure 10: The Traditional Bidirectional LSTM

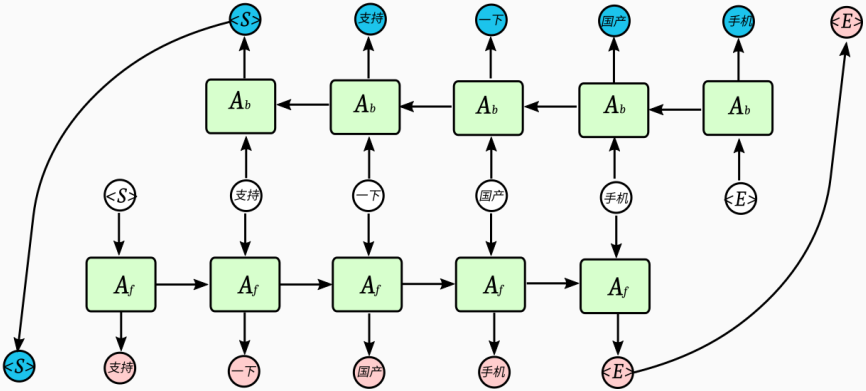

Figure 11: The Modified Bidirectional LSTM

- Similar Posts

$$Score^{1}_{q,p}(q,p) = Sim_{LDA}(q,p) * Sim_{W2V}(q,p) * Sim_{LSTM}(q,p) \quad (13)$$

$$Score^{2}_{q,p}(q,p) = Sim_{LSA}(q,p) * Sim_{W2V}(q,p) * Sim_{LSTM}(q,p) \quad (14)$$

- Comment Candidates

$$Score^{1}_{q,c}(q,c) = Sim_{LSA}(q,c) * Sim_{W2V}(q,c) \quad (15)$$

$$Score^{2}_{q,c}(q,c) = Sim_{LDA}(q,c) * Sim_{W2V}(q,c) \quad (16)$$

Ranking

- TextRank (Words as vertices)
- Pattern-IDF
- Pattern-IDF + TextRank (Sentences as vertices)

Formally, let $G = (V; E)$ be a undirected graph with the set of vertices V and and set of edges E, where E is a subset of $V \times V$. For a given V_i, let $link(V_i)$ be the set of vertices that linked with it. The score of a vertex Vi is define as follow:

$$WS(V_i) = (1 - d) + d * \sum_{j \in link(V_i)} w_{ij} * WSV_j \tag{17}$$

Where d is a damping factor[1]that is usually set to 0.85.

[1]Brin, Sergey, and L. Page. The anatomy of a large-scale hypertextual Web search engine. International Conference on World Wide Web Elsevier Science Publishers B. V. 1998:107-117.

TextRank - Vertices and Edges

- Vertices: each unique word in candidates
- Edges: a co-occurrence relation
- Weighted by: word2vec similarity between two words and the number of their cooccurrences

TextRank - Calculate Iteratively

For N candidates, k words in total, we construct $k \times k$ matrix M.
$M_{ij} = cnt * sim(D_i, D_j)$. Then we compute iteratively

$$R(t+1) = \begin{bmatrix} (1-d)/k \\ (1-d)/k \\ \cdots\cdots\cdots \\ (1-d)/k \end{bmatrix} + d \begin{bmatrix} M_{11} & M_{12} & M_{13} & \cdots & M_{1k} \\ M_{21} & M_{22} & M_{23} & \cdots & M_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{k1} & M_{k2} & M_{k3} & \cdots & M_{kk} \end{bmatrix} R(t)$$

Stop when $|R(t+1) - R(t)| < \epsilon$, $\epsilon = 10^{-7}$.
Here, cnt refers to the number of co-ocurrences within a sentence for D_i and D_j.

Since we get the score $R(D_i)$ for each word D_i in candidates, the score for each comment candidate c is calculated as:

$$Rank_{TextRank}(c) = \frac{\sum_{D_i \in c} R(D_i)}{len(c)} \tag{18}$$

where, $len(c)$ refers to the number of words in comment c.

Pattern-IDF

For word D_i in corresponding comment given word D_j in the post, we define (D_j, D_i) as a pattern.

Inspired by the IDF, we calculate the Pattern-IDF as:

$$PI(D_i|D_j) = 1/\log_2 \frac{count_c(D_i) * count_p(D_j)}{count_{pair}(D_i, D_j)} \tag{19}$$

where $count_c$ refers to the number of word occurring in comments, $count_p$ refers to that in posts, $count_{pair}$ refers to that in post-comment pair. The PI whose $count_{pair}(D_i, D_j)$ less than 3 are eliminated.

Let $X = \frac{count_c(D_i) * count_p(D_j)}{count_{pair}(D_i, D_j)}$, then $X \in [1, \infty)$.

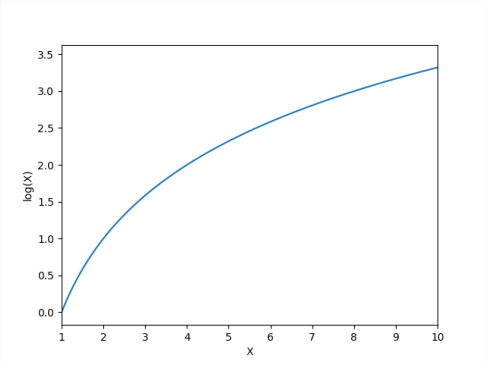

Figure 12: log(X)

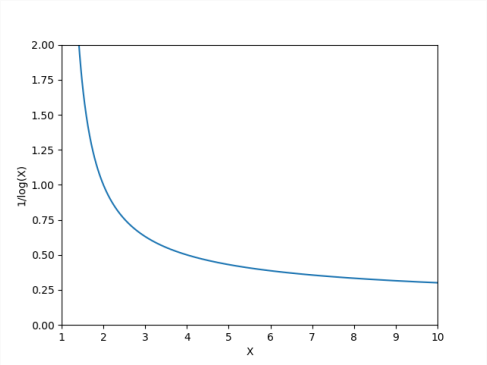

Figure 13: 1/log(x)

PI - Example

Table 2: The example of Pattern-IDF

MajorWord	*MinorWord*	*PI*
中国移动 (China Mobile)	接通 (connect)	0.071725
中国移动	cmcc	0.067261
中国移动	资费 (charges)	0.062408
中国移动	营业厅 (business hall)	0.059949
中国移动	漫游 (roamimg)	0.059234
...	...	...
中国移动	我 (me)	0.028889
中国移动	是 (be)	0.027642
中国移动	的 (of)	0.026346

Table 3: The entropy of Pattern-IDF for each Major Word

MajorWord	H
眼病 (eye disease)	0.889971
丰收年 (harvest year)	0.988191
血浆 (plasma)	1.033668
脊椎动物 (vertebrate)	1.083438
水粉画 (gouache painting)	1.180993
...	...
现在 (now)	9.767768
什么 (what)	10.219045
是 (be)	10.934950

$$PI_{norm}(D_i|D_j) = \frac{PI(D_i|D_j)}{\sum_{i=1}^{n} PI(D_i|D_j)} \tag{20}$$

$$H(D_j) = -\sum_{i=1}^{n} PI_{norm}(D_i|D_j) \log_2 PI_{norm}(D_i|D_j) \tag{21}$$

For each comment c in candidates, given a query (new post) q, we calculate the score by PI as follow:

$$Score_{PI}(q, c) = \frac{\sum_{D_j \in q} \sum_{D_i \in c} PI(D_i|D_j)}{len(c) * len(q)} \tag{22}$$

Then we define rank score as follow:

$$Rank_{PI} = (1 + \frac{Score_{PI}(q, c)}{\max Score_{PI}(q, c)}) * Sim_{W2V}(q, c) * Sim_{LSA}(q, c) \tag{23}$$

In this method, We add each comment sentence in candidates as a vertex in the graph and use sentence Word2Vec similarity as edges between vertices in the graph.

For N candidates, we construct $N \times N$ matrix M. $M_{ij} = SIM_{w2v}(c_i, c_j)$.

At time $t = 0$, We initiate a N-dimension vector P , where N is the number of comment candidates. And each entry of P is defined as the score of Pattern-IDF between the query (new post) q and corresponding comment c_i in candidates:

$$P_i = Score_{PI}(q, c_i) \tag{24}$$

Then we compute iteratively

$$R(t+1) = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \cdots\cdots\cdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} M_{11} & M_{12} & M_{13} & \dots & M_{1N} \\ M_{21} & M_{22} & M_{23} & \dots & M_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{N2} & M_{N3} & \dots & M_{NN} \end{bmatrix} R(t)$$

Stop when $|R(t+1) - R(t)| < \epsilon$, $\epsilon = 10^{-7}$

Finally, we get the score P_i for each comment in candidates.

Experiment

- Nders-C-R5: LDA + Word2Vec + LSTM-Sen2Vec
- Nders-C-R4: LSA + Word2Vec + LSTM-Sen2Vec
- Nders-C-R3: R4 + TextRank (Words as vertices)
- Nders-C-R2: R4 + Pattern-IDF
- Nders-C-R1: R4 + Pattern-IDF + TextRank (Sentences as vertices)

Official Result

Table 4: The official results

Run	Mean nG@1	Mean P+	Mean nERR@10
SG01	0.5867	0.6670	0.7095
splab	0.5080	0.6080	0.6492
Beihang	0.4980	0.5818	0.6105
DeepIntell	0.4323	0.5564	0.5994
Nders-C-R1	0.4593	0.5394	0.5805
Nders-C-R2	0.4743	**0.5497(5th)**	**0.5882(5th)**
Nders-C-R3	0.4647	0.5317	0.5768
Nders-C-R4	**0.4780(4th)**	0.5338	0.5809
Nders-C-R5	0.4550	0.5495	0.5868
R2 vs. R4	↓0.77%	↑2.98%	↑1.26%

Example

```
post:好喜欢这种手绘的画啊，小柚子加油

1 喜欢这种手绘我也要画
2 用手绘板画的吧···
3 天哪好棒！电脑画的还是手绘？
4 很喜欢手绘墙，不过请人画好贵滴。
5 这是用电脑画的还是手绘的？
6 手绘的？拿什么材料画的？
7 我想问你的画是手绘还是电脑画的啊
8 很有爱的画。是用手绘板画的么？
9 这个我也能画，我们以前学习都是手绘的
10 手绘3D画，太牛啦！

post:吃素第一天，坚持住，崔朵拉。
1 是啊，吃素怎么坚持啊
2 请问还在坚持吃素么？
3 吃素的确好可是。坚持不下来哇
4 坚持不住也要坚持。
5 吃素也能发胖么？哦no我坚持吃素结果是骗我的
6 我要坚持不住了。555555
7 坚持不住也得坚持！
8 坚持住，一定要坚持住。
9 我好像坚持不住了……
10 我现在就有点坚持不住了。告诉自己坚持住吧！
```

Figure 14: An example for our result

Questions?