

信息检索 Elasticsearch Pt2

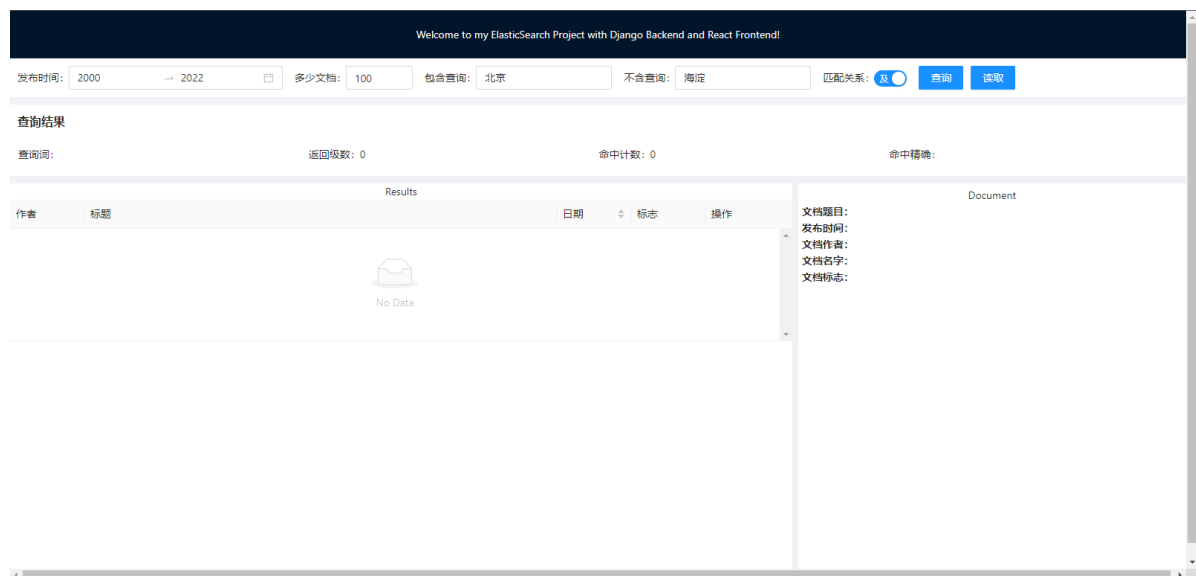
计83 李天勤 2018080106

Abstract

This homework is an extension of the previous Elasticsearch homework. Our goal add more functionality to our IR system by using more techniques that we had learned in the second half of the semester

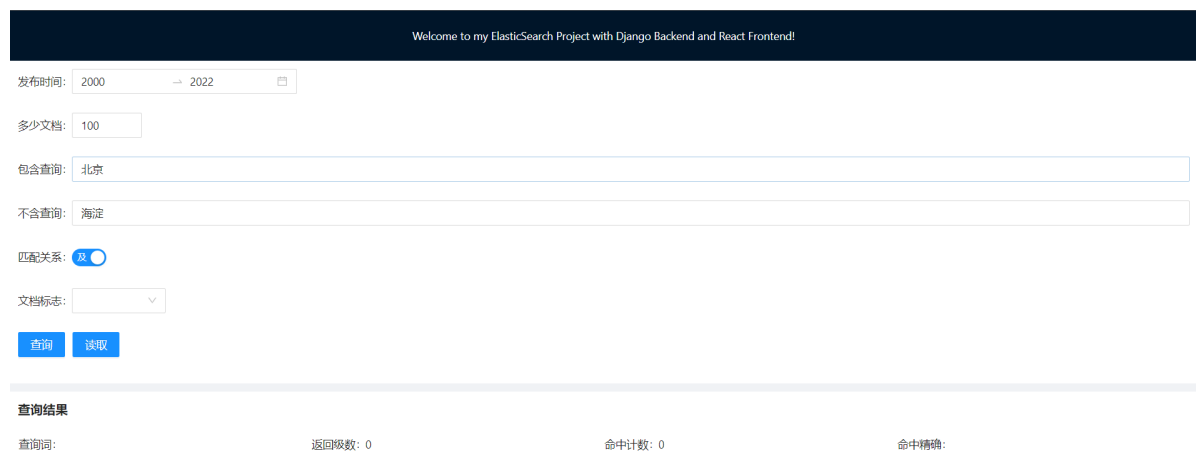
Background

This is what the experiment initially looked like. It only had basic Elasticsearch functionality which is based on Boolean search. It had two search relationships. If the tab displays 及, then the query has an AND relationship. And the 或 describes an OR relationship. This is explained in the documentation



Implementation

I slightly changed the format of the page, and added a couple new functionalities. The first small addition is adding functionality that searches documents by its column. It uses Elasticsearch to match the column sections in the index.



匹配关系：☒ 及

文档标志：

查询

理论

理论专页

光辉理论

理论与实践

查询结果

The document table is the same.

Results							Document		
作者	标题	评分	日期	标志	操作		文档题目:	发布时间:	文档作者:
温红彦 陈晓钟	清华大学学生艺术团赴港演出受欢迎	8.654198	2005-05-07	要闻	阅读		文档名字:		
		8.599146	2001-04-24		阅读				
赵阿娜	清华大学将迎来百年校庆	8.58141	2011-04-09	要闻	阅读				
胡和平	追求卓越的数学家（纪念华罗庚同志诞辰100周年座谈会发言摘编）	8.576933	2010-11-13	综合	阅读				
吴亚明 陈晓星	台湾新竹清华大学隆重庆祝清华创校百年	8.5559225	2011-04-25	要闻	阅读				
		8.555489	2001-04-29		阅读				
		8.549594	2001-04-29		阅读				
袁新文	清华与协和共建医学院	8.547539	2006-09-06	文化新闻	阅读				
赵阿娜	水木清华	8.534168	2011-04-22	要闻	阅读				
< 1 2 > 50 / page									

Word2vec Word Similarity

The following below are the three new functionality that I have added. The first is word similarity. It takes the documents returned by Elasticsearch and does word similarity on the documents.

Word Similarity

Word2Vec分析

输入你自己的query或者它会选择elasticsearch返回的第一个文档

Waiting to press button

No Data

Footer

It splits the request in three sections, and returns the time to read, train, and run.

Word Similarity

Word2Vec分析

输入你自己的query或者它会选择elasticsearch返回的第一个文档

清华

[0.67025] 北航

[0.62691] 清华学

[0.61717] 北

[0.60058] 北师

[0.58442] 校

[0.58338] 海文

[0.56107] 各校

[0.55162] 南开学

[0.54560] 南开

[0.53846] 高校

reading - 39.13464

training - 13.21967

running - 0.43027

The request on the frontend is as such.

```

trainWordSimilarity = () => {
  this.setState({word_sim_list_items: [], word_sim_pending: "Waiting to read", word_sim_loading: true })
  var result = this.state.documents.map(function(a) {return a.id})
  const data = {"query": this.state.query_include, "document_ids": result}
  WordSimilarityReading(data).then(response=>{
    const temp = [{mes: "reading", "time": response.data.results}]
    this.setState({ word_sim_reading_time: response.data.results, word_sim_list_items: temp, word_sim_pending: "Waiting to train"})
    console.log(response.data);
  }).then(training => {
    const data2 = {"task": "train"};
    WordSimilarityTraining(data2).then(response=>{
      const temp = [{mes: "reading", "time": this.state.word_sim_reading_time }, {"mes": "training", "time": response.data.results}]
      this.setState({ word_sim_training_time: response.data.results, word_sim_list_items: temp, word_sim_pending: "Waiting to run"})
      console.log(response.data);
    }).then(running => {
      if (this.state.documents.length === 0) {
        this.setState({ word_sim_pending: "Waiting to press button", word_sim_loading: false, word_sim_list_items: []})
        message.error("没有语料库, 请查询")
        return;
      }
      const data3 = {"task": "run", "query": this.state.word_sim_chosen_word === "" ? this.state.query_include : this.state.word_sim_chosen_word }
      WordSimilarityRunning(data3).then(response=>{
        console.log(response.data);
        if (response.data.results.results === "keyerror") {
          message.error("这个词不在词典里面, 重新选")
          this.setState({word_sim_running_time: "", word_sim_list_data: [], word_sim_pending: false, word_sim_loading: false});
          return;
        }
        const temp = [{mes: "reading", "time": this.state.word_sim_reading_time}, {"mes": "training", "time": this.state.word_sim_training_time }, {"mes": "run", "time": response.data.results.time}]
        this.setState({ word_sim_list_items: temp})
        this.setState({word_sim_running_time: response.data.results.time, word_sim_list_data: response.data.results.results, word_sim_pending: false, word_sim_loading: false})
      }).catch(error=>(message.error(error)))
    }).catch(error=>(message.error(error)))
  }).catch(error=>(message.error(error)))
}

```

And the python backend is simply implements the functionality of word2vec. For example, the train function is as such. The implementations for the following are similar.

```

def word_sim_train():
    """train corpus based on parameters"""
    start = time.time()
    lines = []
    with open("data.jsonl") as f:
        lines = f.read().splitlines()
    text_corpus = json.loads(lines[0])
    text_corpus = [text.split() for text in text_corpus]

    params = {
    }

    # train model
    model = Word2Vec(sentences=text_corpus, **params)
    model.save(f"sg{params['sg']}_hs{params['hs']}_win{params['window']}_size{params['vector_size']}.model")

    end = time.time()
    spent_time = end - start
    return '%.5f'%spent_time


```

Doc2vec Document Similarity

I used Gensim's Doc2vec to implement document similarity. You can enter your own document or query, or it will use the first result returned by elasticsearch. In word2vec, you train to find word vectors and then run similarity queries between words. In doc2vec, you tag your text and you also get tag vectors. Then, after doc2vec training you can use the same vector arithmetic's to run similarity queries on author tags, which document are most similar to the one being queried.

Document Similarity
Doc2Vec分析

Waiting to press button


No Data

Footer

It returns the top 5 results

输入你自己的文档或者它会选择elasticsearch返回的第一个文档

[0.32835] [288874] 本报驻印尼记者 董力一年前的 5 月 2 7 日，印尼爪哇岛中部的日惹特别自治区发生了里氏 5．9 级地震和 3 次里氏 5 级左右的余震，造成巨大人员伤亡和财产损失。在救灾和重建过程中，印尼政府得到了国际社会的大力援助，印尼各地的华人社团也积极伸出援手，受到当地政府和人民的赞许。班图尔县是受灾最严重的地方，有 4 万多人死亡，占死亡总人数的近 7 0 %；近 1 4 万幢房屋完全倒塌或严重损毁，占倒塌房屋总数的 4 7 %。地震一周年之际，记者来到了班图尔县，了解当地民众重建家园的情况。前往班图尔县的路上，还可以看到不少地震留下的痕迹。不过，在班图尔县，一幢幢新房子已经代替了去年的残垣断壁，很多房子一看就是新修的。当然，半新半旧的房子也有很多，不少人仍在废墟上修葺着自家的房子。大街上人来人往，各类车辆川流不息。让人对班图尔两年重建计划的按时完成充满信心。班图尔县政府发言人巴姆邦介绍说，班图尔县一年来的重建工作成果显著：卫生医疗服务已经恢复正常；超过 7 0 % 的学校恢复教学；道路和水利等基础设施也基本完全修复。巴姆邦说，班图尔县灾后重建的速度之所以这么快，首先在于民众能够互相帮助。大家自发组织起来，共同出力，一家一家地修复房屋；其次，离不开政府以及国际社会的大力支持。当地政府为每户灾民补贴 1 5 0 0 万印尼盾（相当于 1．5 万元人民币），联合国“爪哇重建统筹基金会”也为灾民提供大量资金，用于房屋重建，中国等国还派遣了医疗队救治在地震中受伤的灾民，还有很多国家也提供了援助物资和资金。另外，印尼各地的民间组织和社团，尤其是华人社团也为重建做了许多贡献。日惹福清公会主席陈世祥和日惹印尼华裔总会秘书游平茂介绍说，日惹以及印尼全国的华人社团都纷纷伸出援手，参加救灾。地震发生后不久，日惹的华人社团就筹集了大批的食品、饮用水、毛毯、帐篷和药品，并组织人员前往一个个受灾的村庄，把物资送到灾民手上，解了他们的燃眉之急。日惹的华人社团还积极协助中国政府派来的医疗队，为医疗队工作的顺利进行提供了便利条件。灾后重建开始后，华人又捐赠了大量的资金，用于修复学校、资助灾民恢复生活等。在走访中记者看到，日惹福清公会捐建 1 2 亿印尼盾修建的贾拉坎小学校舍宽敞明亮（见上图，本报记者董力摄），是印尼华人社团捐建的近 2 0 所学校中规模最大的一所。学校的围墙和校门上写着一个红艳艳的“融”字。校长德斯莉曼女士介绍说，这个“融”字代表了华人和其他民族兄弟般的融洽关系。同时，这个字也可以提醒学生，长大以后要促进与华人的团结。陈世详说，学校正式开学时，日惹特区领导人还在讲话中称赞华人社团的无私帮助。班图尔县是整个日惹地区灾后重建的缩影。在 2 6 日举行的地震一周周年纪念活动上，印尼总统苏西洛称赞日惹和爪哇地区在灾后重建方面取得的成就，他无不骄傲地说：“以前外国人总是问印尼政府什么时候才能结束善后工作，而现在，他们问的是政府如此迅速地进行灾后重建有什么秘诀。”苏西洛的话从另一个侧面表明，无论是印尼政府还是当地百姓，都对灾区未来充满了信心。（本报印尼日惹电）

- reading - 40.23807
- training - 3.79919
- running - 0.95861

Lsi Document Similarity

Similar to document similarity, but still using LSI.

Lsi Similarity

输入你自己的文档或者它会选择elasticsearch返回的第一个文档

No Data

Footer

Waiting to press button

Lsi Similarity

输入你自己的文档或者它会选择elasticsearch返回的第一个文档

[0.9999999] [2189880] 本报 香港 5 月 6 日电 记者 温红彦、陈晓钟 报道：“清韵华情”满载着促进清华大学与香港交流的使命，满载着清华学子的殷殷祝福，今晚溢满了美丽的香江。在清华大学校庆 94 周年之际，清华大学艺术团一行 68 位师生来到香港，与香港青年欢度五四青年节，为香港各界朋友和清华校友献上了一场场精彩纷呈、热情奔放的民族歌曲、舞蹈、器乐和综合文艺节目。清华大学党委书记陈希教授专程赴香港出席了艺术团的演出活动，代表母校向香港各界人士、向广大的清华校友送上节日问候。清华大学艺术团这次是应香港清华联会的邀请到港演出的，昨晚和今晚在香港理工大学的赛马会综艺馆演出，明天还将在香港葵涌葵锦秋中学为全港中学生演出。“清韵华情”是清华大学学生艺术团的品牌型综合文艺演出。中联办教育科技部副部长王国力说，这次演出反映了清华大学弘扬民族文化、倡导高雅艺术的宗旨，展现了浓浓的中华情和校友与母校之间的亲情，同时也展示了清华学生的艺术修养。据了解，这次演出的所有组织工作都是由清华大学在港校友完成的。香港清华联合会干事长方方先生介绍说，这次访问从去年就开始筹备，在港的清华校友为了这次演出投入了大量的精力、物力和人力。“清韵华情”也是清华大学与香港各界交流的桥梁。清华大学党委副书记张再兴教授介绍说，这次演出既是向香港各界汇报清华大学的艺术教育成果，也是答谢香港各界对清华大学建设的帮助和支持。今晚，清华学子们熟悉的蒙民伟学长也观看了演出。演出一结束，学生们纷纷围上去与蒙先生合影：“天天进出清华大学的‘蒙民伟楼’，今天终于见到了您！”香港理工大学博士生王树晓、研究生肖慧看完演出告诉记者，节目非常精彩，反映了内地大学生昂扬向上的精神风貌，希望两地的大学生经常有这样的交流活动。

- reading - 40.20096
- training - 52.76355
- running - 0.41847

Analysis

We can see from the results that Doc2Vec and LSI to compare each method.

Using Elasticsearch, we queried the corpus and got back 10000 results

发布时间: 2000 → 2022

多少文档: 10000

包含查询: 清华大学

不含查询: Ex 海淀

匹配关系: 取

The result with the highest score is this Elasticsearch query is

查询结果						
查询词: 北京但不包含海淀(及)		返回极数: 10000		命中计数: 10000		命中精确: gte
Results						Document
作者	标题	评分	日期	标志	操作	文档题目: 《中国大博览》北京分册出版发行 发布时间: 2007-03-03 文档作者: 智文 文档名字: deta11@record=2742&channelId=200703&searchword=&sortfield= 文档标志: 国际要闻
智文	《中国大博览》北京分册出版发行	4.1907825	2007-03-03	国际要闻	阅读	本报北京 3月1日讯 在北京 紫柳密处 迎接 2008年奥运会之际, 21世纪中国大型对外宣传画册——《中国大博览》北京分册正式出版发行。《中国大博览》北京分册为大8开精装本,分上下两册,中英文对照,共计46万字,2000余幅图片;由魅力北京、辉煌北京、经贸北京、风采北京、旅游北京、投资北京、畅想北京、明珠北京、信息北京9部分组成,全方位、多角度反映了近年来首都北京发展的新成就、新风貌、新举措,充分展现了老北京的辉煌历史、新北京的迷人魅力和未来北京的美好愿景,具有较强的权威性、实用性和收藏价值。中共中央政治局委员、北京市委书记刘淇为该分册作序,主编为北京市市长王岐山。(智文)
	关于表彰2005年度区域经济五十强二三产业先进镇、二三产业先进村招商引资先进集体、文明富裕村等的决定	4.1759167	2006-03-01	公告	阅读	
		4.1672716	2002-12-07		阅读	
		4.1637135	2001-02-21		阅读	
钟文	将真实的北京展示给世界(五环漫笔)	4.1616807	2008-07-09	奥运特刊	阅读	
阎晓明 王建新	燕京啤酒成为北京奥运会赞助商 (2008北京奥运之窗)	4.1604757	2005-08-11	体育	阅读	

Then, to compare the methods, we can look at the results given. The algorithms and methods used for this extension are the same as the homework for 6 and 7, and the conclusions are pretty much the same. LSI is a count based model where similar terms have same counts for different documents. Then dimensions of this count matrix is reduced using SVD. For both the models similarity can be calculated using cosine similarity. Word2vec is a prediction based model, for example, the given the vector of a word predict the context word vectors (such as skipgram method). When utilizing a small window count, Doc2vec organizes results by terms that are 相似, while LSI organizes results by 相关. Training an LSI system takes much more time than Doc2vec on its basic window of 5, and with 5 epochs.

Compared to Doc2vec and LSI, Elasticsearch is much faster, and cheaper, and handles only pure and simple keyword searches. Thus, when handling simple keyword searches such as 清华 or 北京, Elasticsearch gives us a much better results when we search for documents with simple queries.

Thus, when handling larger document based similarity searches (larger queries), it is better to use Doc2Vec or LSI methods.

Improvements

Based on this experiment, and throughout this class, it is easy to see how easy to hard it is to make a good IR system. There are a lot of things to consider. If I have more time to work on this experiment, I would like to have implemented more features, and tried more information retrieval methods, such as BERT, and do a deeper comparison between these methods and find a way to display the similarities and differences. I would also like to have improved the interface.

Thanks!