《信息检索》第三次作业

计83李天勤2018080106 George Li

分别寻找任意一个针对英文的Stemmer和 Lemmatizer（也可以用课堂PPT上推荐的），任意选择风格不一致的三个文章小片段，分别做stemming和lemmatization, 观察结果并做比较，进一步地，对其对信息检索可能造成的影响进行分析。

Find any Stemmer and Lemmatizer for English (you can also use the ones recommended in the classroom PPT), choose any three small pieces of articles with inconsistent styles, do stemming and lemmatization respectively, observe the results and compare them, and further compare them Analysis of the possible impact of information retrieval.

I used the Stemmer/Lemmatizer recommended on the class PowerPoint: https://snowballstem.org/demo.html

Article 1: Holding RIPK1 On the Ubiquitin Leash in TNFR1 Signaling by Nieves Peltzer, Maurice Darding, and Henning Walczak

> Current Knowledge of TNFR1 Signaling
>
> The process of ubiquitination, that is, the covalent attachment of ubiquitin to target proteins (Box 1), is involved in various biological processes such as the regulation of protein stability, cell cycle, receptor trafficking, and cell signaling [1,2]. The role of ubiquitination has been extensively studied in tumor necrosis factor (TNF) signaling. TNF belongs to the TNF superfamily of cytokines that play an important role in inflammation, innate and adaptive immunity, cell death, and cancer. These cytokines can bind to their cognate receptors, the members of the TNF receptor superfamily (TNFR-SF), induce their trimerization, and initiate signaling pathways, resulting in proinflammatory gene activation. TNFR1 belongs to a subset of the TNFR-SF termed death receptors (DRs) that are characterized by the presence of an intracellular death domain (DD). Apart from gene induction, these receptors are also able to induce programmed cell death by apoptosis and, as has more recently emerged, by necroptosis, which is regulated form of necrosis.

Result of the demo

> the process of ubiquitin, that is, the coval attach of ubiquitin to target protein (box 1), is involv in various biolog process such as the regul of protein stabil, cell cycl, receptor traffick, and cell signal [1,2]. the role of ubiquitin has been extens studi in tumor necrosi factor (tnf) signal. tnf belong to the tnf superfamili of cytokin that play an import role in inflamm, innat and adapt immun, cell death, and cancer. these cytokin can bind to their cognat receptor, the member of the tnf receptor superfamili (tnfr-sf), induc their trimer, and initi signal pathway, result in proinflammatori gene activ. tnfr1 belong to a subset of the tnfr-sf term death receptor (drs) that are character by the presenc of an intracellular death domain (dd). apart from gene induct, these receptor are also abl to induc program cell death by apoptosi and, as has more recent emerg, by necroptosi, which is regul form of necrosi

We can see that words like proteins, cytokins, and pathways, were successfully lemmatized into protein, cytokin, and pathway. There were a lot of words that had an affix of s, and they were reduced. In terms of stemming, words such as ubiquitination, stability, cycle, trafficking, inflammation were reduced into ubiquitin, stabil, cycl, traffick, and inflamm. Biological terms, such as cytokin, ubuiquitin, and protein are already considered roots, and therefrore were not reduced nay further. Biology contains lots of acronyms, and they were simply reduced to lower letters.

Article 2: P4 Programming Protocol Independent Packet Processors

This article is about the P4 programming language, a method of using programmable switches to create our networks.

> 3. A PROGRAMMING LANGUAGE
>
> We use the abstract forwarding model to define a language to express how a switch is to be configured and how packets are to be processed. This paper's main goal is to propose the P4 programming language. However, we recognize that many languages are possible, and they will likely share the common characteristics we describe here. For example, the language needs a way to express how the parser is programmed so that the parser knows what packet formats to expect; hence a programmer needs a way to declare what header types are possible. As an example, the programmer could specify the format of an IPv4 header and what headers may legally follow the IP header. This motivates defining parsing in P4 by declaring legal header types. Similarly, the programmer needs to express how packet headers are to be processed. For example, TTL fields must be decremented and tested, new tunnel headers may need to be added, and checksums may need to be computed. This motivates P4's use of an imperative control flow program to describe header field processing using the declared header types and a primitive set of actions.

Result

> we use the abstract forward model to defin a languag to express how a switch is to be configur and how packet are to be process. this paper's main goal is to propos the p4 program languag. howev, we recogn that mani languag are possibl, and they will like share the common characterist we describ here. for exampl, the languag need a way to express how the parser is program so that the parser know what packet format to expect; henc a programm need a way to declar what header type are possibl. as an exampl, the programm could specifi the format of an ipv4 header and what header may legal follow the ip header. this motiv defin pars in p4 by declar legal header type. similar, the programm need to express how packet header are to be process. for exampl, ttl field must be decrement and test, new tunnel header may need to be ad, and checksum may need to be comput. this motiv p4's use of an imper control flow program to describ header field process use the declar header type and a primit set of action

There isn't too many interesting things going on in this article. Words such as packets, characteristics, and types are lemmatized/stemmed to packet, characterist, and type. There are the removal of derivational affixes such as -ive in primative. However some words like decremented, I was confused why it wasn't reduced to decre, as ment is a affix as well as ed, and decre is a root of the word decrement. Words like added were reduced to ad, which might be confusing to informational retreival systems because I believe it can also represent words such as advertisement, which often uses ad for short.

Article 3: Jittor: a novel deep learning framework with meta-operators and unified graph execution

This is a deep learning framework that I used in my computer graphics course to recognize numbers.

> Dynamic binding allows users to take full advantage of the underlying hardware performance when using a scripting language, but it has a problem: all operations are statically compiled, making optimizations such as operator fusion difficult; this important optimization technique combines multiple operators into one operator, so that intermediate results do not need to be stored. Dynamic binding with a scripting language cannot use this

> optimization. For example, the user may need to calculate d = ab + c, where a, b, c are tensors. First, the scripting interpreter executes tmp = TensorMul(a,b) and then executes d = TensorAdd(temporary,c). If we could compile the whole expression, rather than applying operators one by one, we could execute d = TensorMulAndAdd(a,b,c) directly without the need for temporary storage. This is significant, as on modern processors, memory access is often much slower than calculation. However, we cannot guess what combinations of operators the user may require, and static compilation of all possible combinations is obviously infeasible. To solve this problem, we may use JIT compilation technology to dynamically compile and optimize the operators that the user needs.

Result

> dynam bind allow user to take full advantag of the under hardwar perform when use a script languag, but it has a problem: all oper are static compil, make optim such as oper fusion difficult; this import optim techniqu combin multipl oper into one oper, so that intermedi result do not need to be store. dynam bind with a script languag cannot use this optim. for exampl, the user may need to calcul d = ab + c, where a, b, c are tensor. first, the script interpret execut tmp = tensormul(a,b) and then execut d = tensoradd(tmp,c). if we could compil the whole express, rather than appli oper one by one, we could execut d = tensormulandadd(a,b,c) direct without the need for temporari storag. this is signific, as on modern processor, memori access is often much slower than calcul. howev, we cannot guess what combin of oper the user may requir, and static compil of all possibl combin is obvious infeas. to solv this problem, we may use jit compil technolog to dynam compil and optim the oper that the user need

Here, we see that there are some mathematical expressions involved in the article, sch as d = ab +c, which remained unchanged. Formula names however, such as tmp = tensormul(a,b) become lowercase. I was interested to see if the algorithm would lemmatize words contained inside the formulas, such as d = tensoradd(tmp,c). So I slightly changed the initial input to tensoradd(temporary, c), and it resulted in d = tensoradd(temporari,c). Thus, this shows that stemmatization might be a problem when it comes to documents containing code, with variables as words.