

| 자랑 게시판 | >

[스압] 자연어 파싱하고 감정분석 하기 (학습이란 이런것)



윤 챗봇 입문자 1:1 채팅
2020.06.04. 16:16 조회 333

댓글 63 URL 복사

이 카페를 보거나, 다른 봇 커뮤니티 채팅방을 보면 "학습봇" 에 대하여 많은 얘기가 거론됩니다.

저는 이 학습봇이란 말을 듣고, '세상에, 고 쪼그만 안드로이드에 머신러닝을 탑재한다고?!' 란 신선한 충격을 받았습니
다.

하지만 언제나 현실은 냉정한 법이죠.

실제 대부분의 기능은 채팅을 기록하거나 특정 명령어에 대한 문구를 추가합니다.
가장 최근 글의 소스를 봤습니다.

무작정 "랜덤"으로 수집한 글을 가지고, "랜덤" 적인 말을 "랜덤" 한 시점에 보냅니다.



“
그건 학습이 아니라 "저장"과 "명령어 추가"일 뿐이야!
”

실제 머신러닝을 설명하기엔 준비해야 할 것도 많고, 이론마저 복잡하기 때문에 그건 차차 알아보기로 합시다.

이번 글에서는 사람이 얼마나 부정적인지 감정을 분석하여, 위로와 진정의 말을 건내는 답변을 보내봅니다.

1. 감정이 얼마나 많은데?

맞습니다. 사람의 감정은 이루 말할 것 없이 복잡하며 종잡기 힘듭니다.
그게 쉬웠으면 심리학자같은 것도 없을 테고, 전부 기계가 감정을 분석해서 일상에 조언을 주겠죠.

희망을 품는, 기쁜, 즐거운, 설레는, 행복한, 등. 많은 항목이지만 이것을 "긍정적인 감정" 이라고 볼 수 있습니다.
화나는, 짜증나는, 슬픈, 등. 이것들은 반대로 "부정적인 감정" 이라고 볼 수 있습니다.

우리는 부정적인 감정 (-1) ~ 긍정적인 감정 (1) 사이 어느정도인지 수치로 분석할 것이며, 이에 따른 동작을 추가할 수
있어야 합니다

2. 분석, 네가 그걸 해?



아, 물론 제가 안 합니다. 제가 할 수 있었으면 이미 대기업에 있었지, 이럴 시간이 어디에 있겠나요. 우린 대기업의 기술을 가져다 쓸 겁니다. 그것도 아주아주 세계적으로 큰 구글의 기술을 말이죠.



Google Cloud Platform

구글의 클라우드 서비스 플랫폼. 일명 GCP에 있는 Natural Language 를 사용하겠습니다.
먼저, 구글 계정이 있지만 해당 플랫폼을 사용해 본 적 없다는 가정하에 진행하겠습니다.

프로젝트 관리자

프로젝트 관리자에 들어가서 오른쪽에 프로젝트 만들기를 누르면 새 프로젝트 창이 뜹니다.
만들어 줍니다. 기존에 사용하던 프로젝트가 있으신 분은 사용해도 상관 없습니다.

Google Cloud Platform

🔍 제품 찾기

새 프로젝트

⚠️

projects 할당량이 21개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제 하세요. [자세히 알아보기](#)

[MANAGE QUOTAS](#)

프로젝트 이름 *

Kakao-Bot

?

프로젝트 ID: kakao-bot-279301입니다. 나중에 변경할 수 없습니다. [수정](#)

위치 *

🗺️ 조직 없음

[찾아보기](#)

상위 조직 또는 폴더

만들기

취소

API 사용 설정

Google Cloud Platform에 Cloud Natural Language API용으로 애플리케이션 등록

Google Cloud Platform에서 애플리케이션을 관리하고 API 사용량을 모니터링할 수 있습니다.

애플리케이션을 등록할 프로젝트 선택
한 개의 프로젝트를 사용하여 모든 애플리케이션을 관리하거나, 애플리케이션별로 다른 프로젝트를 만들 수 있습니다.

Kakao-Bot

계속

근데 이상한 서비스 계정을 만들으라고 하니, API 정보를 만들기 위하여 다음 주소로 이동합니다.

API 키가 생성됩니다.

3. REST API 란?

{ REST:API }

지금 점심을 먹어서 노곤한 상태입니다.

장황한 설명은 하기 귀찮으니, 간단하게 말하여 http 통신으로 인해 원하는 기능을 수행한 결과를 가져온다고 하죠. 즉, 우리는 http 통신을 사용해 텍스트에서 감정분석을 할 예정입니다.

구글 자연어 서비스는 이 REST API 를 지원합니다. 그를 위하여 API 키를 발급한 것이고요.

순서는 간단하게 다음과 같습니다.

1. 문장을 구글 NL 서비스에 전달
2. 구글 NL 서비스가 분석함
3. 분석한 결과를 전달 받음.



데이터를 받는 나의 모습

우리는 앱에서 사용할 것이니 jsoup 를 사용하겠지만, 일단 되는지 확인을 해보도록 하죠. 어차피 제가 대표로 되는지 확인해 드릴테니, 여러분들은 구경만 하시면 됩니다.

API DOCS 를 보면, POST 요청을 보내면서 데이터들이 필요하다고 합니다.
우선, 지원하는 API 들은 다음과 같습니다.

Methods	
<code>analyzeEntities</code>	Finds named entities (currently proper names and common nouns) in the text along with entity types, salience, mentions for each entity, and other properties.
<code>analyzeEntitySentiment</code>	Finds entities, similar to <code>AnalyzeEntities</code> in the text and analyzes sentiment associated with each entity and its mentions.
<code>analyzeSentiment</code>	Analyzes the sentiment of the provided text.
<code>analyzeSyntax</code>	Analyzes the syntax of the text and provides sentence boundaries and tokenization along with part of speech tags, dependency trees, and other properties.
<code>annotateText</code>	A convenience method that provides all the features that <code>analyzeSentiment</code> , <code>analyzeEntities</code> , and <code>analyzeSyntax</code> provide in one call.
<code>classifyText</code>	Classifies a document into categories.

영어를 모르겠대구요? 그런건 파파고나 시키십쇼.



설명해라 인간

예. 설명하겠습니다.

`analyzeEntities` : 이름 및 명사를 찾는 기능입니다.
예를 들어, "아버지가 방에 들어가신다." 에서 "아버지" 를 뜻합니다.

`analyzeEntitySentiment`: 대충 위에서 설명한 기능과 아래에서 설명할 기능의 복합기능인데, 한국어는 지원하지 않습니다.

`analyzeSentiment`: 문자의 정서를 분석하는 기능입니다.

analyzeSentiment: 문장의 감정을 분석하는 기능입니다.

analyzeSyntax: 문장을 토큰으로 나누고, 해당 문장에 대한 음성 정보를 세부적으로 표현합니다.

- 형태소 분석이라 보시면 될 것 같습니다.

annotateText: 모든 분석 기능들을 통합한 기능입니다. 이거 한 번 호출하면 다른 기능들을 여러번 호출하지 않아도 됩니다.

classifyText: classify 라는 소리가 정확히 뭔진 모르겠지만, 카테고리를 분류하는 기능입니다. 하지만 한국어는 지원하지 않습니다.

이제 여기서 우리는 한 번 호출해서 모든 정보를 알아낼 수 있는 annotateText를 사용할 것입니다.
요청 Url 은 다음과 같습니다.

“
<https://language.googleapis.com/v1/documents:annotateText>
”

그리고 파라미터로 아까 발급받은 API KEY를 전달해 주어야 합니다.

<https://language.googleapis.com/v1/documents:annotateText?key=발급받은Key>

위 주소로 POST 요청을 보내야 합니다.
보내는 데이터 정보는 아래와 같습니다.

JSON representation	
<pre>{ "document": { object(Document) }, "features": { object(Features) }, "encodingType": enum(EncodingType) }</pre>	
Fields	
document	object(Document) Input document.
features	object(Features) The enabled features.
encodingType	enum(EncodingType) The encoding type used by the API to calculate offsets.

JSON 타입으로, document. 즉 보내는 문장에 대한 정보.
features 어떠한 기능들을 사용할 것인지.
encodingType 문자열 인코딩 타입입니다.

document 정보 형식은 다음과 같습니다.

JSON representation

```
{
  "type": enum(Type),
  "language": string,

  // Union field source can be only one of the following:
  "content": string,
  "gcsContentUri": string
  // End of list of possible types for union field source.
}
```

일단 백건이 붙여일문이라고 하죠. 아니, 백문이 붙여일건인가?
간단하게 표현해 드리자면 아래와 같겠네요.
위 정보들을 토대로 보내야 할 JSON 데이터를 만듭니다.

```
{
  "document": {
    "type": "PLAIN_TEXT",
    "language": "ko",
    "content": "나는 그거 절대 싫어."
  },
  "features": {
    "extractSyntax": true,
    "extractEntities": true,
    "extractDocumentSentiment": true,
    "extractEntitySentiment": false,
    "classifyText": false
  },
  "encodingType": "UTF8"
}
```


이 요청에 대한 답은 아래와 같았습니다.
정보가 많아서 그런지 좀 길게 나왔네요.

```
1 {
2   "sentences": [
3     {
4       "text": {
5         "content": "나는 그거 절대 싫어.",
6         "beginOffset": 0
7       },
8       "sentiment": {
9         "magnitude": 0.7,
10        "score": -0.7
11      }
12    }
13  ],
14  "tokens": [
15    {
16      "text": {
17        "content": "나",
18        "beginOffset": 0
19      },
20      "partOfSpeech": {
21        "tag": "PRON",
22        "aspect": "ASPECT_UNKNOWN",
23        "case": "CASE_UNKNOWN",
24        "form": "FORM_UNKNOWN",
25        "gender": "GENDER_UNKNOWN",
26        "mood": "MOOD_UNKNOWN",
27        "number": "NUMBER_UNKNOWN",
28        "person": "PERSON_UNKNOWN",
29        "proper": "NOT_PROPER",
30        "reciprocity": "RECIPROCITY_UNKNOWN",
31        "tense": "TENSE_UNKNOWN",
32        "voice": "VOICE_UNKNOWN"
33      },
34      "dependencyEdge": {
35        "headTokenIndex": 4,
36        "label": "NSUBJ"
37      },
38      "lemma": "나"
39    },
40    {
41      "text": {
42        "content": "는",
43        "beginOffset": 3
44      },
45      "partOfSpeech": {
46        "tag": "PRT",
47        "aspect": "ASPECT_UNKNOWN",
48        "case": "NOMINATIVE",
49        "form": "FORM_UNKNOWN",
50        "gender": "GENDER_UNKNOWN",
51        "mood": "MOOD_UNKNOWN",
52        "number": "NUMBER_UNKNOWN",
53        "person": "PERSON_UNKNOWN",
54        "proper": "NOT_PROPER",
55        "reciprocity": "RECIPROCITY_UNKNOWN",
56        "tense": "TENSE_UNKNOWN",
57        "voice": "VOICE_UNKNOWN"
58      },
59      "dependencyEdge": {
60        "headTokenIndex": 0,
61        "label": "PRT"
62      },
63      "lemma": "는"
```

```

63     "lemma": "은"
64 },
65 {
66     "text": {
67         "content": "그거",
68         "beginOffset": 7
69     },
70     "partOfSpeech": {
71         "tag": "ADV",
72         "aspect": "ASPECT_UNKNOWN",
73         "case": "CASE_UNKNOWN",
74         "form": "FORM_UNKNOWN",
75         "gender": "GENDER_UNKNOWN",
76         "mood": "MOOD_UNKNOWN",
77         "number": "NUMBER_UNKNOWN",
78         "person": "PERSON_UNKNOWN",
79         "proper": "NOT_PROPER",
80         "reciprocity": "RECIPROCITY_UNKNOWN",
81         "tense": "TENSE_UNKNOWN",
82         "voice": "VOICE_UNKNOWN"
83     },
84     "dependencyEdge": {
85         "headTokenIndex": 4,
86         "label": "ADVMOD"
87     },
88     "lemma": "그거"
89 },
90 {
91     "text": {
92         "content": "절대",
93         "beginOffset": 14
94     },
95     "partOfSpeech": {
96         "tag": "ADV",
97         "aspect": "ASPECT_UNKNOWN",
98         "case": "CASE_UNKNOWN",
99         "form": "FORM_UNKNOWN",
100        "gender": "GENDER_UNKNOWN",
101        "mood": "MOOD_UNKNOWN",
102        "number": "NUMBER_UNKNOWN",
103        "person": "PERSON_UNKNOWN",
104        "proper": "NOT_PROPER",
105        "reciprocity": "RECIPROCITY_UNKNOWN",
106        "tense": "TENSE_UNKNOWN",
107        "voice": "VOICE_UNKNOWN"
108    },
109    "dependencyEdge": {
110        "headTokenIndex": 4,
111        "label": "ADVMOD"
112    },
113    "lemma": "절대"
114 },
115 {
116     "text": {
117         "content": "싫어",
118         "beginOffset": 21
119     },
120     "partOfSpeech": {
121         "tag": "ADJ",
122         "aspect": "ASPECT_UNKNOWN",
123         "case": "CASE_UNKNOWN",
124         "form": "FINAL_ENDING",
125         "gender": "GENDER_UNKNOWN",
126         "mood": "INDICATIVE",
127         "number": "NUMBER_UNKNOWN",
128         "person": "PERSON_UNKNOWN",
129         "proper": "NOT_PROPER",
130         "reciprocity": "RECIPROCITY_UNKNOWN",
131         "tense": "TENSE_UNKNOWN",

```

```

132     "voice": "VOICE_UNKNOWN"
133   },
134   "dependencyEdge": {
135     "headTokenIndex": 4,
136     "label": "ROOT"
137   },
138   "lemma": "싫어"
139 },
140 {
141   "text": {
142     "content": ".",
143     "beginOffset": 27
144   },
145   "partOfSpeech": {
146     "tag": "PUNCT",
147     "aspect": "ASPECT_UNKNOWN",
148     "case": "CASE_UNKNOWN",
149     "form": "FORM_UNKNOWN",
150     "gender": "GENDER_UNKNOWN",
151     "mood": "MOOD_UNKNOWN",
152     "number": "NUMBER_UNKNOWN",
153     "person": "PERSON_UNKNOWN",
154     "proper": "NOT_PROPER",
155     "reciprocity": "RECIPROCITY_UNKNOWN",
156     "tense": "TENSE_UNKNOWN",
157     "voice": "VOICE_UNKNOWN"
158   },
159   "dependencyEdge": {
160     "headTokenIndex": 4,
161     "label": "p"
162   },
163   "lemma": "."
164 }
165 ],
166 "entities": [],
167 "documentSentiment": {
168   "magnitude": 0.7,
169   "score": -0.7
170 },
171 "language": "ko",
172 "categories": []
173 }

```

우리는 결과의 sentences를 볼것입니다. sentiment 의 score 가 감정의 수치거든요.
-1.0 이면 완전 부정적, 1.0 이면 완전 긍정적입니다.

자, 그러면 "나는 그거 절대 싫어." 라는 말의 스코어가 -0.7로, 좀 많이 부정적인 감정이네요.
한 -0.4 이하이면 반응을 하는게 좋을 것 같습니다.

4. 분석의 기준 문장.

카카오톡은 그 특성상, 한 텍스트 안에 제대로 되는 문장을 받기가 힘듭니다.

-> 아
-> 근데 사실
-> 이거 개노잼이니까
-> 따라하지 마셈.

위와 같은 방식으로 한 문장을 여러 채팅으로 나눠보내는 경우가 많으며

-> ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ
-> ㅎㅎㅎㅎ
-> ㄴㄴㄴ

와 같은 방식으로 정상적으로 분석 불가능한 문장을 보내는 경우도 있습니다.
또한, 제가 할 것은 명령어로 인한 감정분석이 아니라, 실제 채팅중을 분석할 것이므로 모든 채팅을 검사해야 합니다.



얼마 뒤 핸드폰 램 상황

간단하게 조건을 몇 가지 정리해 봅시다.
이 조건은 언제든지 바꿀 수 있습니다.

1. 문장을 구성하는 한글 [가-힣] 만 검사 요소에 포함시킨다.
2. 문장은 5글자 이상이다.
3. 같은 글자가 문장에 50% 이상 있으면 정상적 문장으로 취급하지 않는다.

위 조건에 부합하는 문장들만 감정분석을 시도하도록 합니다.
그러면, 함수를 하나 구현해야겠죠?

```
1 function canAnalyticText(text) {
2     if ( typeof text !== "string" ) return false;
3
4     trimText = text.replace(/\s/g, "");
5     korText = trimText.match(/[가-힣]/g)
6
7     if ( korText ) {
8         korTextLen = korText.length;
9         check = {};
10
11         for ( t of korText ) {
12             if ( check[t] ) {
13                 check[t] += 1;
14             } else {
15                 check[t] = 1;
16             }
17         }
18
19         keys = Object.keys(check);
20
21         for ( k of keys ) {
22             if ( check[k] >= (korTextLen / 2) ) {
23                 return false;
24             }
25         }
26     }
```

```

25     }
26
27     return true;
28 }
29
30

```

Colored by Color Scripter

구현했습니다.

간단한 테스트 결과, 생각보다 괜찮은 것 같기도 합니다.

```

> canAnalyticText('안녕하세요?')
true
> canAnalyticText('안안안요?')
false
> canAnalyticText('안안')
false
> canAnalyticText('코로코로스모')
false
> canAnalyticText('cand')
false
>

```

5. 슬슬 읽기가 힘들어.

일단 여기까지 읽으셨다니, 정말 제 글을 열심히 읽어주셔서 감사하다는 뜻을 전합니다.
저도 쓰기가 힘듭니다. 하지만 이제 슬슬 마무리입니다.



이제 정상적인 문장이라고 판단을 했으면, 감정분석을 해야죠.

Jsoup 를 사용하여 post 전송 할겁니다.

```

1  apiUrl = "https://language.googleapis.com/v1/documents:annotateText";
2  key = "[Your API Key]";
3  function textAnalysis(text) {
4      data = {
5          "document": {
6              "type": "PLAIN_TEXT",

```



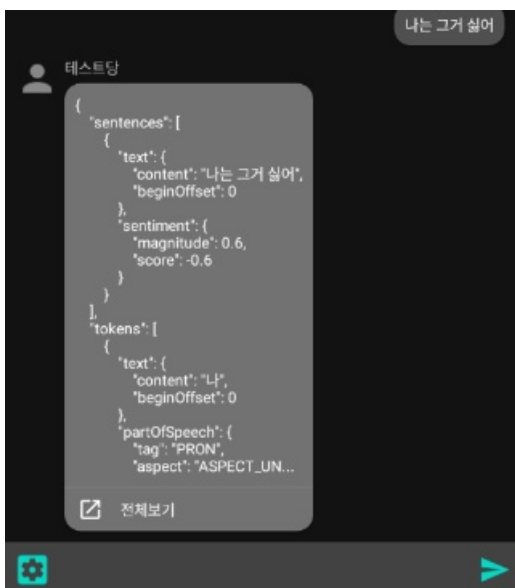
```

7      "language": "ko",
8      "content": text
9  },
10  "features": {
11      "extractSyntax": true,
12      "extractEntities": true,
13      "extractDocumentSentiment": true,
14      "extractEntitySentiment": false,
15      "classifyText": false
16  },
17  "encodingType": "UTF8"
18  };
19
20  result = Jsoup.connect(apiUrl + "?key=" + key)
21      .userAgent("Mozilla")
22      .ignoreContentType(true)
23      .header("Content-Type", "application/json; charset=UTF-8")
24      .requestBody(JSON.stringify(data))
25      .post();
26  r = JSON.parse(result.body().text());
27  return r;
28  }
29

```

Colored by Color Scripter

감정분석 결과를 받는 코드입니다.



정상적으로 받는 것을 확인하였습니다.

이젠 진짜 진짜 마지막 단계인, -0.4 이하의 부정적인 감정을 검사하는 것입니다.
그리고 귀찮으니 "힘내." 한 마디 보내주는 걸로 때우죠.

```

1  function isNegative(anal) {
2      sumScore = 0;
3      if ( anal && anal.sentences ) {
4          s = anal.sentences;
5          for ( sentence of s ) {
6              if ( sentence.sentiment ) {
7                  sumScore += sentence.sentiment.score;
8              }
9          }
10         if ( (sumScore / s.length) <= -0.4 ) {
11             return true;
12         }
13     }
14 }

```

```

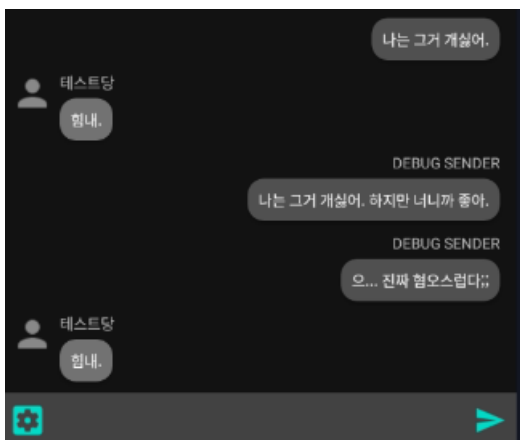
13     }
14
15     return false;
16 }
17
18 function response(room, msg, sender, isGroupChat, replier, imageDB, packageName) {
19     if ( canAnalyticText(msg) ) {
20         r = textAnalysis(msg);
21
22         if ( isNegative(r) ) {
23             replier.reply("힘내.");
24         }
25     }
26 }
27

```

Colored by Color Scripter

됐습니다.

감정 분석은 "문장마다" 해서 배열로 주기 때문에, 모든 문장의 전체 평균을 구하여 -0.4 이하인지 검사합니다.



성공입니다.

2번째 문장의 경우

"나는 그거 개싫어." : -0.6

"하지만 너니까 좋아." : 0.9

평균 0.15 이므로, 부정적인 문장이 아니어서 "힘내." 란 텍스트를 보내지 않았습니다.
전체 소스 올리겠습니다.

```

1  Jsoup = org.jsoup.Jsoup;
2
3  function canAnalyticText(text) {
4      if ( typeof text !== "string" ) return false;
5
6      trimText = text.replace(/\s/g, "");
7      korText = trimText.match(/[가-힣]/g)
8
9      if ( korText ) {
10         korTextLen = korText.length;
11         check = {};
12
13         for ( t of korText ) {
14             if ( check[t] ) {
15                 check[t] += 1;
16             } else {
17                 check[t] = 1;
18             }
19         }
20
21         let sum = 0;
22         for ( t in check ) {
23             sum += check[t];
24         }
25
26         let avg = sum / korTextLen;
27
28         return avg < -0.4 ? -0.4 : avg > 0.4 ? 0.4 : avg;
29     }
30     return false;
31 }
32

```

```

16         } else {
17             check[t] = 1;
18         }
19     }
20
21     keys = Object.keys(check);
22
23     for ( k of keys ) {
24         if ( check[k] >= (korTextLen / 2) ) {
25             return false;
26         }
27     }
28
29     return true;
30 }
31
32 return false;
33 }
34
35
36 apiUrl = "https://language.googleapis.com/v1/documents:annotateText";
37 key = "[Your API Key]";
38 function textAnalysis(text) {
39     data = {
40         "document": {
41             "type": "PLAIN_TEXT",
42             "language": "ko",
43             "content": text
44         },
45         "features": {
46             "extractSyntax": true,
47             "extractEntities": true,
48             "extractDocumentSentiment": true,
49             "extractEntitySentiment": false,
50             "classifyText": false
51         },
52         "encodingType": "UTF8"
53     };
54
55     result = Jsoup.connect(apiUrl + "?key=" + key)
56         .userAgent("Mozilla")
57         .ignoreContentType(true)
58         .header("Content-Type", "application/json;charset=UTF-8")
59         .requestBody(JSON.stringify(data))
60         .post();
61     r = JSON.parse(result.body().text());
62     return r;
63 }
64
65 function isNegative(anal) {
66     sumScore = 0;
67     if ( anal && anal.sentences ) {
68         s = anal.sentences;
69         for ( sentence of s ) {
70             if ( sentence.sentiment ) {
71                 sumScore += sentence.sentiment.score;
72             }
73         }
74         if ( (sumScore / s.length) <= -0.4 ) {
75             return true;
76         }
77     }
78
79     return false;
80 }
81
82 function response(room, msg, sender, isGroupChat, replier, imageDB, packageName) {
83     if ( canAnalyticText(msg) ) {
84         r = textAnalysis(msg);

```

```

85
86         if ( isNegative(r) ) {
87             replier.reply("힘내.");
88         }
89     }
90 }
91

```

Colored by Color Scripter

6. 진짜 끝

쓰는 저도 매우 힘들었습니다.

읽는 여러분은 얼마나 힘들지 감히 상상하기도 힘들지만 읽어주신 여러분들 모두 감사드립니다.

아, 이게 왜 성과 게시판에 있냐구요?

저는 Jsoup 도 처음 사용해본 것이며, 사실 이 구글의 Natural Language 도 처음 사용해본 것입니다.
그러니 처음 도전해서 성과를 내었다 볼 수 있죠.



윤님의 게시글 더보기 >

❤ 좋아요 13 💬 댓글 63

🔗 공유 | 신고

댓글 등록순 최신순 ↻

댓글알림 ☐



사과님

감정분석이라 좋네요 잘봤습니다.

2020.06.04. 16:17 답글 쓰기



윤 작성자

벌써요?! 되게 빠르네요

2020.06.04. 16:18 답글 쓰기



산하

ㄷㄷ 작성하시느라 수고하셨습니다

(제 봇에는 이미 탑재되어있던 기능((((음음)

2020.06.04. 16:18 답글 쓰기



윤 작성자

역시 저 같은 초보는 감히 그림자도 못 쫓겠네요

2020.06.04. 16:21 답글 쓰기



산하

윤 당신은 이미 제 위에 있답니다☆



윤 작성자

이제 보니 sentiment 들의 평균을 구하는 것 보다 documentSentiment 를 사용하면 평균을 구하는 작업을 하지 않아도 되네요!

2020.06.04. 16:52 답글쓰기



성빈

우와...

2020.06.04. 16:54 답글쓰기



윤 작성자

봐주셔서 감사합니다. 토할 것 같아요.

2020.06.04. 16:54 답글쓰기



nolbo

크으

2020.06.04. 17:43 답글쓰기



윤 작성자

봐주셔서 감사합니다.

2020.06.04. 17:53 답글쓰기



Ametrine

필력 좋으시네요!



2020.06.04. 17:51 답글쓰기



윤 작성자

아유... 그냥 주절주절 거리는 것입니다. 좋게 봐주셔서 감사합니다.

2020.06.04. 17:53 답글쓰기



Dark Tornado

조사 분리를 /*열라게 대충*/ 구현한 이유가 사라지는 게시글이었던 거시예오.

2020.06.04. 18:52 답글쓰기



윤 작성자

이제 주석처리를 하면 되는 거시예오

2020.06.04. 18:53 답글쓰기



재승

와..

말이 안나온다

2020.06.04. 19:08 답글쓰기



윤 작성자

말을 많이하면 줄기 힘듭니다

2020.06.04. 19:11 답글쓰기



야옹

말이 안나오면 소가 나와야 읊음

2020.06.04. 21:24 답글쓰기



재승

야옹 ㅣ 이해 해 버렸어

2020.06.04. 21:24 답글쓰기



야옹

	재승 당신의 나이는?	2020.06.04. 21:57	답글쓰기	
	재승 야옹 흑염룡/*흑우*/이 돌아 나는 나이입니다	2020.06.04. 21:58	답글쓰기	⋮
	야옹 재승 오호 그렇다면 당신은 20대군	2020.06.04. 22:00	답글쓰기	⋮
	재승 야옹 ㅊ라 아직 10대인데	2020.06.04. 22:00	답글쓰기	⋮
	야옹 재승 중2인가요	2020.06.04. 22:02	답글쓰기	⋮
	재승 야옹 ㄸ 아직 중2중2 하진 않아요요	2020.06.04. 22:03	답글쓰기	⋮
	야옹 재승 그렇다면 말투를 보아하니 찜민이는 아닌 듯 하고 중2중2하지는 않으니 40대겠군(?)	2020.06.04. 22:04	답글쓰기	⋮
	야옹 재승 중1인가요!	2020.06.04. 22:04	답글쓰기	⋮
	재승 야옹 ㅊ라 어째서 결론이?	2020.06.04. 22:04	답글쓰기	⋮
	재승 야옹 중yee요	2020.06.04. 22:05	답글쓰기	⋮
	야옹 재승 보시오 var 나이=age 찜민이=6<나이<13 중2=나이==15 그러므로 (6<나이<13)!=0&&(나이<15) 그러므로 나이=40	2020.06.04. 22:07	답글쓰기	⋮
	재승 야옹 ㅊ라	2020.06.04. 22:08	답글쓰기	⋮
	재승 야옹 논리적인 헛소리군	2020.06.04. 22:08	답글쓰기	⋮
	야옹 재승 Test for (논리적인 것 같이 보이는) mistyped as (논리적인)?	2020.06.04. 22:11	답글쓰기	⋮
	재승 야옹 ㅊ라 ㅌㅌ			⋮

2020.06.04. 22:12 답글쓰기



야옹
재승 나에게서 도망치지마요(?)

2020.06.04. 22:17 답글쓰기



야옹
잘 쓰시네요 그나저나 ㄸㄸ

2020.06.04. 21:24 답글쓰기



윤 작성자
칭찬해 주셔서 감사합니다

2020.06.04. 21:29 답글쓰기



야옹
윤 저한테 한 소리음음

2020.06.04. 21:58 답글쓰기



야옹
야옹 장난입니다

2020.06.04. 21:58 답글쓰기



재승
'그'나 '저'나 ㄸ

2020.06.04. 22:12 답글쓰기



야옹
재승 그 뒤의 말을 알아맞추시오(5점)

2020.06.04. 22:17 답글쓰기



재승
야옹 나에게서 도망치세요

2020.06.04. 22:19 답글쓰기



Error0918
와우..

2020.06.04. 23:25 답글쓰기



AlphaDo
봐주셔서 감사합니다.

2020.06.07. 00:30 답글쓰기



윤 작성자
잘 봤습니다. 글 잘 쓰시네요 ㄸㄸ

2020.06.07. 00:32 답글쓰기



jomin398
구글 NLP 기간제 무료아니던가요...?

2020.06.07. 09:56 답글쓰기



윤 작성자
기간제 무료는 Auto ML NL 입니다.
글에서 사용하고 있는 것은 NL API 입니다.

자세한 가격책정 정책은 다음 두 곳을 참고하시면 됩니다.

<https://cloud.google.com/natural-language/automl/pricing?hl=ko>

<https://cloud.google.com/natural-language/pricing?hl=ko>

2020.06.10. 10:01 답글쓰기



franknoh
textAnalysis(msg)
값이 계속
[object Object]
로 나와요

2020.06.10. 12:07 답글 쓰기



윤 작성자

해당 함수는 객체값을 객체로 반환합니다. 그러니 해당 객체를 문자열로 확인하고 싶으시면 JSON.stringify를 사용하여 먼저 문자열로 만드셔야 합니다.

2020.06.10. 12:23 답글 쓰기



franknoh

윤 감사합니다

2020.06.10. 12:28 답글 쓰기



책사주세요

이렇게 뜨는데 아무리 고쳐보려고 해도 잘 안 됩니다..ㅠㅠ 61번째 줄은
`r = JSON.parse(result.body().text());`
여기입니다 도와주실 수 있을까요?



2020.06.24. 14:47 답글 쓰기



윤 작성자

`result.body().text()` 의 값을 출력해 보세요

2020.06.24. 15:11 답글 쓰기



책사주세요

윤 그 값을 출력하라는게 어떤 의미인가요? JSON화시키지 말고 r에 저것만 저장해서 하라는 말씀이신가요..?ㅠㅠ 초보라 의미 파악을 잘 못해서...

2020.06.24. 15:16 답글 쓰기



윤 작성자

책사주세요 JSON parse 하기 전에, 제대로 된 JSON 형식인지 확인해봐야 합니다.

`Log.d(result.body().text())` 로 로그 결과값을 확인하든

`Api.reply(result.body().text(), room)` 으로 확인하든 어떠한 방법으로든 `result.body().text()` 값을 확인해 보세요

2020.06.24. 15:17 답글 쓰기



책사주세요

윤 메신저봇 어플로도 로그 확인할 수 있는 방법이 있나요...? 시도해보는데 안되는 것 같아서...

2020.06.24. 15:33 답글 쓰기



윤 작성자

책사주세요 매봇 가이드 잘 보면 있습니다.

2020.06.24. 15:34 답글 쓰기



책사주세요

윤 올려주신 코드 추가하고 실행시킨 후 로그 기록을 보려고 했는데 로그 기록이 나오는 곳에 아무것도 쓰여져있지 않아요..

2020.06.24. 15:44 답글 쓰기



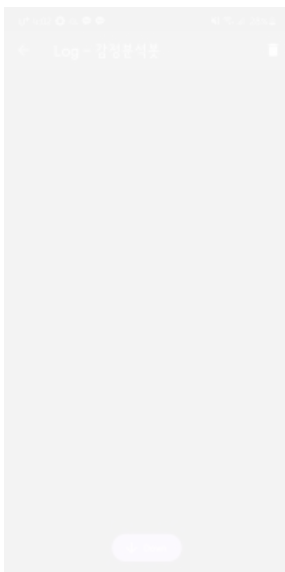
윤 작성자
책사주세요



2020.06.24. 16:01 답글쓰기



책사주세요
윤 네 ㅠㅠ 그거 확인했는데... 안 뜨네요



2020.06.24. 16:02 답글쓰기



윤 작성자
책사주세요 그대로 실행했을 때 로그는 안 뜨고 에러는 그대로인가요?

2020.06.24. 16:15 답글쓰기



책사주세요
윤 네 맞습니다ㅠㅠ 에러가 난 줄 바로 밑에 Log.d(result.body().text()) 추가해서 실행시켜봤지만 에러는 그대로 뜨고 로그는 아무것도 안 나와요 ㅠㅠ 바쁘실텐데 감사합니다

2020.06.24. 16:17 답글쓰기



윤 작성자
책사주세요 위에 추가해야 합니다.

2020.06.24. 16:17 답글쓰기



책사주세요
윤 앗 위에 추가하니깐 뜨는군요... 근데 어마어마하게 길게 나와요 로그가...

2020.06.24. 16:22 답글쓰기



윤 작성자
책사주세요 그걸 해석해 봐야죠

2020.06.24. 16:36 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'자랑 게시판' 게시판 글

이 게시판 새글 구독하기 ☐

[스압] 핸드폰 안 만지고 새로운 내용 적용하기 🤖 [27]	윤	2020.06.05.
구글시트로 공지 시스템 구축했어요 🤖 [6]	첸징	2020.06.04.
[스압] 자연어 파싱하고 감정분석 하기 (학습이란 이런것) 🤖 [63]	윤	2020.06.04.
드디어 멜론 성공했어요! 🤖 [13]	팀메우	2020.06.03.
ㄹㅇ 개뽈짓 🤖 [11]	replier	2020.06.02.

1 2 3

전체보기

이 카페 인기글



아키네이터

안녕하십니까.

JSR

♡0 💬8

로코가 뭐길래 사람들이 환장하는걸까?


쿨칠

♡1 💬18

오픈이발 쓰는사람 계정 영정먹이는 방법

Kiri

♡0 💬14



카카오톡이 이런걸 추천하다니

ACTIVE

(수정5)카카오톡 오픈채팅 공식 봇관련 정

[봇드게임] 세븐 다이스 v1.1.0

ZUMP

♡0 💬2

카톡 정지에 관해서

리스트

♡0 💬15

getSchoolFood

도미 doami2005

♡0 💬5

1 2 3 4 5