# 腾讯云文智自然语言处理

## 产品白皮书

[2015.11.30]

[V1.0]

## 【版权声明】



## 【商标声明】



## 【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

# 目录



# 1 前言

自然语言处理最早在 20 世纪 50 年代的美国萌芽，在半个多世纪里有了不少的发展。近几年，互联网非结构化数据的迅速增长， 管理非结构化文本数据，挖掘其中的价值成为了一种趋势。目前，自然语言处理市场的主要公司包括 3M、苹果、杜比系统、谷歌、惠普、IBM、微软 、、NetBase Solutions、SAS 软件研究所、Verint Systems 等。据估计，到 2020 年，全球自然语言处理市场价值将到到 134 亿美元。从 2015 年起，复合年增长率为 18.4%。

# 2 产品简介

文智自然语言处理 ,基于并行计算系统和分布式爬虫平台 ,结合独特的语义分析技术, 一站式满足用户自然语言分析、转码、抽取、全网数据抓取等需求，用户能够基于平台对外提供的 OpenAPI 实现搜索、推荐、舆情、挖掘等语义分析应用，也能够通过与我们合作定制产品特色的语义分析解决方案。

## 2.1 使用好处

### 2.1.1 高效精准分析文本数据

传统人工对文本的标注分析工作效率极低，且成本高，使用文智 API，可在短时内分析海量文本，且文智 API 以千亿级的互联网语料数据为基础，其准确度能帮助更好的挖掘出文本价值。

### 2.1.2 降低运用自然语言处理技术的成本

自然语言研究技术门槛高，用户自主研发及维护成本高。使用文智 API，用户无需考虑

自主研发的技术难题，同时大大降低成本。

## 2.2 产品特点

### 2.2.1 调用简单方便

API 调用简单方便，不限制编程语言种类，短短几行代码即可获取分析结果。

### 2.2.2 语义接口丰富全面

API 语义接口丰富全面，且不断迭代优化。

## 2.3 核心优势

### 2.3.1 分析精准

十年专注语义分析，千亿级互联网语料，众多腾讯产品应用经验，为文智 API 的精准效果打下了基础。

### 2.3.2 服务高效

服务支撑高效稳定，同时配备完善及时的开发者支持

### 2.3.3 接口全面

一站解决中文语义分析需求，集合词法、句法、篇章、下载模块，提供分词\命名实体识别、情感分析、主题分类、关键词提取、同义词、纠错、下载等多种 API，且更多 API 也将不断推出。

## 2.4 产品功能

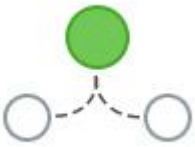
### 2.4.1 分词/命名实体识别 API

提供智能分词（基本词和短语）、词性标注、命名实体识别功能。专业的团队对数据、模型、程序进行迭代更新以保证效果的不断提升。用户只需简单的调用相关 API 接口即可获取到所需结果，无需担心诸如新词发现、歧义消除、调用性能等词法分析难题。词法分析已经为应用宝搜索、微信公共账号搜索等业务提供支持，均取得了良好的效果。

### 2.4.2 情感分析 API

为有情感分析需求的产品提供的服务。该服务能够对信息进行情感上的正向、负向及中性进行评价。在舆情监控、话题监督、口碑分析等商业分析领域有非常重要的应用价值。

### 2.4.3 主题分类 API



为用户提供自动文本分类服务，平台已对文本分类的模型算法进行了封装，用户只需提供待分类的文本数据，而不必关注具体的实现，通过平台就能得到提供文本的所属类别。目前平台能识别类别囊括了求职招聘、影视、音乐、健康养生、财经、广告推广、犯罪、政治等 40 多个类别，且算法支持快速迭代更新已有类别及增加新类别。

### 2.4.4 关键词提取 API

基于关键词抽取平台，为用户实现诸如新闻内容关键词自动提取、评论关键词提取等提供基础服务。支持用户自定义词典，提高在垂直领

域的抽取效果。目前已经接入的业务包括腾讯新闻客户端、手机腾讯网等。

### 2.4.5 同义词 API

为用户提供同义词查询服务，搜索团队通过全网数据挖掘出海量同义词，并持续对数据、模型等进行迭代更新，保证同义词的效果始终与时俱进。用户也可以通过提供产品专有的数据，与我们合作打造专属的同义词库。同义词服务作为搜索引擎检索串理解的基本功能，目前已经应用在视频、音乐、应用宝、群搜、商圈等数百个产品中。

### 2.4.6 纠错 API

能够实现对短文本的自动纠错功能，长文本的自动纠错也即将推出。用户只需要提供业务数据和日志，无需关注技术细节和更新流程，就可以享受到业务自身定制的纠错服务，甚至不提供业务数据，享受通用的纠错服务。 目前已经接入的业务包括音乐、视频、应用宝、云搜等， 评测效果均好于竞品。

### 2.4.7 转码 API

分为两大类：网页转码和网页名片。网页转码将在 PC 机上展示的二维页面转换为适合在手机等移动端设备上展示的一维页面，方便用户在移动端阅读。网页名片将页面简化为主体图片、标题、摘要的组合，以“卡片”的形式展示给大众，适合做页面的分享、收藏、推广等。用户只需要提交网页的 url ，就能获取我们的转码服务，方便、快捷。当前，网页转码已为公司 QQ、qzone、微云、微博、正文吧等平台提供服务。

### 2.4.8 下载抽取 API

基于分布式爬虫系统，用户只需提供一个 url 即可轻松完成数据抓取，也可与下载团队合作打造专有的定向抓取服务。分布式爬虫系统通过对全网 url 进行精准调度、智能压力挖掘、自适应页面更新周期预测，可以实现自动路由、url 作弊识别、智能主题抓取等功能。水平的架构设计使得系统可以进行任意的扩展，同时结合公司海量运营的经验，在系统监控、运营告警等方面都不断进行完善使得系统可以稳定高效运行。

## 3 应用场景

文智自然语言处理的应用场景颇为广泛。只要有大量文本的地方，都可以是文智的应用场景。语义搜索，个性化推荐推荐、舆情监控，问答机器人是其中几大应用场景。另外诸如，机器自动撰写新闻，品牌口碑分析，公关舆情监控，社交网络社群聚类，垃圾邮件识别等等功能也都可由文智 API 来辅助实现。

## 4 案例

### 4.1 案例：游戏舆情项目 Wetest


该产品的主要功能是监控游戏类 app 的口碑，跟踪热点评论，洞察用户心声。而文智为其提供的情感分析 API 支撑起了其舆情功能。该项目当前对文智情感分析 API 的日调用量超过 5000 万。

### 4.2 案例：手 Q 阅读圈

该产品使用了文智文本分类 API，主要用于给文章自动打标签，而基于该标签再给用户做读物推荐。

## 5 快速上手

### 5.1 产品购买

介绍页入口：http://www.qcloud.com/product/nlp.html

购买页入口：http://manage.qcloud.com/shoppingcart/shop.php?tab=wenzhi

### 5.2 使用指南

腾讯云文智中文语义平台以 SDK 模块的方式提供服务，多种编程语言都可以轻松使用。在正式使用之前，您需要首先在腾讯云上注册文智账号。

这里将以一个简单的情感分析任务为例，介绍腾讯云 sdk 文智模块的使用。

首先请在腾讯云官方 sdk 下载地址
https://github.com/QcloudApi/qcloudapi-sdk-php
下载或更新最新版本的 sdk（本次以 php-sdk 为例）； 修改 demo.php 文件，修改点如下:

a) SecretId，SecretKey 改为自己腾讯云上相应的值，这里查看：
http://manage.qcloud.com/capi/capiManage.php

b ) $package=array('offset'=>0, 'limit'=>3); 改为：

$package = array("content"=>"李亚鹏挺王菲：加油！孩儿他娘。");

说明：这是文智情感分析接口的参数。

c ) $a=$cvm->DescribeInstances($package); 改为：

$a = $wenzhi->TextSentiment($package);

说明：这是文智模块的相关接口，具体请查看接口列表：

http://www.qcloud.com/wiki/API 说明文档

d ) 其他所有地方的$cvm 改为$wenzhi，即替换为文智模块。

修改后的 demo.php 如下：

```
<?php
error_reporting(E_ALL ^ E_NOTICE);
require_once './src/QcloudApi/QcloudApi.php';

$config = array(
                'SecretId'       => '你在腾讯云上的 SecretId',
                'SecretKey'      => '你在腾讯云上的 SecretKey',
                'RequestMethod'  => 'POST',
                'DefaultRegion'  => 'gz');

$wenzhi = QcloudApi::load(QcloudApi::MODULE_WENZHI, $config);

$package = array("content"=>"双万兆服务器就是好，只是内存小点");

$a = $wenzhi->TextSentiment($package);
```

```
if ($a === false) {
    $error = $wenzhi->getError();
    echo "Error code:" . $error->getCode() . ".\n";
    echo "message:" . $error->getMessage() . ".\n";
    echo "ext:" . var_export($error->getExt(), true) . ".\n";
} else {
    var_dump($a);
}

echo "\nRequest :" . $wenzhi->getLastRequest();
echo "\nResponse :" . $wenzhi->getLastResponse();
echo "\n";
```