Technical Report for iDMlib

Eric

April 20, 2010

Document History		
Date	Author	Content
2010-04-019	Eric Xing	Write an intial format

Abstract

This document presents the technical report for the project iDMLib in iZENEsoft. iDMLib aims to build a core library that integrates the various basic data mining algorithms and components that are developed in iZENEsoft to maximize the resusability.

Contents

1 Introduction **3**

2 Duplicate Detection **3**

2.1 API usage 3

3 Key Phrase Extraction **3**

3.1 API usage 3

4 Name Entity Classification **3**

4.1 API usage 3

5 Document Classification **4**

5.1 API usage 4

1 Introduction

2 Duplicate Detection

2.1 API usage

3 Key Phrase Extraction

3.1 API usage

4 Name Entity Classification

Name Entity Classification module is used to classify a string(supposed to be a noun phrase) to following classes:

- PEOP, represent for people, for examples: 黄昆, 哈里逊, 组织代表, 克里斯蒂安八世
- ORG, represent for ogranization, for examples: 北师大, 北京地质学院, 黑手党
- LOC, represent for location, for examples: 奥地利共和国, 九龙塘铁路站
- OTHER, noun phrase but not name entity, for examples: 巴士路线, 政治风气
- NOISE, non-noun phrase, for examples: 直线传播, 分析和解决

Note that this module is not a module of Name Enity Recognition, which can recognize name entity from a text. The difference is that the input string of Name Entity Classification is supposed to be a noun phrase, while the input string of Name Entity Recognition may be a sentence, paragrahp or a whole article.

4.1 API usage

The usage of this module is very simple, some test cases like `test/nec/xxx.cpp` can be refered.

- Following code snippet demonstrate the usage of the module training.

```
vector<NameEntity> entities;
// load entities
...

// set model path
string path = "model_path";
NameEntityManager neMgr(path);

// training
neMgr.train(entities);
```

- Following code snippet demonstrate the usage of the module prediction.

```
// set model path
string path = "model_path";
NameEntityManager neMgr(path);
// load nec models
neMgr.loadModels();

vector<NameEntity> entities;
NameEntity entity;
// load entities
...

// prediction
neMgr.predict(entities);
// or
// neMgr.predict(entity);
```

5 Document Classification

5.1 API usage