システムディスクリプション

記入日：2012 年 8 月 8 日
対象トラック： オープントラック
システム名： SMT を用いた誤り検出（＋前置詞，動詞一致）
チーム名： NAIST
性能： ドライラン R=0.270 P=0.418 F=0.328 (前置詞，動詞一致含まず)
フォーマルラン R=0.288 P=0.484 F=0.361

1. 概要

本システムは統計的機械翻訳の手法を用いて誤り訂正を行ない，システムが訂正を行なった箇所を誤りとして検出する．また，ワークショップのトラックにある前置詞，動詞の一致の検出結果を足すことができる．

2. 実行方法

実行環境・使用言語：Perl

実行方法：

0. Moses，GIZA++，SRILM をインストール

1. 学習者コーパス（***.incor と***.corr）とテストコーパスを言語モデル用コーパスを準備

学習者コーパスは学習者の文とその正解が対になっていれば何でも良い．

データは小さいが KJ コーパスで良い

言語モデル用コーパスは***.corr のファイルでも良い

(1-1. preprocessing.sh によってトークナイズ，小文字化を行なう．)

2. lm.sh で言語モデルを作成

```
./lm.sh lm_example.txt
```

3. train.sh で訂正モデルを学習

```
./train.sh cm_example /***/***/ lm_example.txt.lm
```

4. test.sh でテストデータに対して訂正を行なう

```
./test.sh test_example.txt
```

(4-1. postprocessing.sh によって detokenize する)

5. edcw_tag.pl によって<gen>タグを付けて誤り箇所の検出を行なう

```
perl edcw_tgag.pl test_example.txt test_example.txt.res
```

(5-1. merge_result_detect.pl で前置詞，動詞の一致の結果のマージを行なう)

perl merge_result_detect.pl –a test_example.txt.res –p prp_example.txt.res –v vagr_example.txt.res

(注：shellscript 内部で一部ツールのコマンドを絶対パスで指定して呼び出している箇所があるため，PATH を指定して書き換えるもしくは自分の環境に合わせて書き換えが必要である．)

3. 使用データ，ツール，辞書

ツール名：

moses (http://www.statmt.org/moses/)

GIZA++ (http://giza-pp.googlecode.com/files/giza-pp-v1.0.7.tar.gz)

SRILM (http://www.speech.sri.com/projects/srilm/download.html)

任意の学習者コーパス（学習者の文と添削文が対になっているもの）

4. システムの詳細

処理の流れは次の通りである

（1）moses を使って訂正モデルを学習する

（2）moses を使って誤り訂正を行なう

（3）（2）で元のテストデータと訂正した結果を比べて異なっている箇所を誤りとして検出しタグの付与を行なう

（3-1）前置詞や動詞の一致の結果とのマージを行なう

5. セールスポイント

大規模な学習者コーパスがあれば，そこから自動で訂正モデルを作成して誤りの訂正を行なうことができ，誤り訂正結果を見ることで検出も可能である．