01 tokenization

Kazuki Yabe

September 2019

1 Segmenter

1.1 Segmenters

1.1.1 Segmenter from the slide

This segmenter simple add new line after "?", "!", or ".".

1.1.2 NLTK segmenter

This segmenter uses nltk.tokenize.sent_tokenizer for sentence segmentation.

1.2 Quantitative

I manually looked at the first 200 examples, and calculated the accuracy for both segmenters. NLTK segmenter performed slightly better than the segmenter from the slide by 1%, where NLTK got 100% accuracy and The slide's segmenter got 99.5% accuracy. The only difference was that the slide's segmenter wrongly segmented a quate with a period inside quatation marks.

1.3 Qualitative

NLTK segmenter performs better in segmenting URLs, quotations, abreviations, and decimals.

Types	NLTK	Slide
URL	`http://c2.com/cgi/wiki?WikiHistory`	`http://c2.`
Quatations	"mangingibig ng karunungan**."**	"mangingibig ng karunungan**.**
Abreviations	Osmeña **Jr."** at lolo nina ...	Osmeña **Jr.**
Decimal Numbers	hanggang **26.5**% ...	hanggang **26.**

URL, quotation with period eg "hello." Jr. 26.5 (decimal number)

2 Tokenizer

2.1 Implementation

/01_Tokenisation/maxmatch.py

2.2 Instruction

```
python maxmatch.py
```

The above command run maxmatch on test data which is extracted from test.conllu. Use ">file_name" to save the result to a desired file.

```
python maxmatch_evaluation.py
```

The above command run WER on results of maxmatch.py. Use ">file_name" to save the result to a desired file.

2.3 Evaluation

WER score is calculated for evaluation of the maxmatch algorithm. The score is 8.4%. In general, if a word is not in dictionary, then the tokenizer separates that word into some miscellaneous words that the tokenizer recognizes.

For example, the algorithm had a problem in identifying names of organizations, like "幸福の科学". It parsed it as "幸福" "の" "科学," by treating the name as noun-particle-noun.

Since there are many homograph