Part of Speech Tagger Result

Tongzhou Hai
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

1 Result Table

		NN			VB		
	acc	prec	rec	f	prec	rec	f
wsj.sec24	95.57%	93.56%	95.65%	94.59%	96.06%	90.93%	93.43%
gweb-answers	87.24%	78.39%	86.10%	82.06%	87.03%	76.22%	81.27%
gweb-emial	87.25%	82.67%	76.99%	79.73%	89.11%	83.72%	86.33%
gweb-newsgroup	89.56%	83.83%	87.01%	85.39%	93.23%	84.63%	88.72%
gweb-reviews	89.77%	87.81%	87.49%	87.56%	91.94%	87.39%	89.61%
gweb-weblogs	92.72%	87.65%	91.60%	89.58%	94.10%	86.56%	90.17%

Table 1: Figures from first 5000 sentences of training file

		NN			VB		
	acc	prec	rec	f	prec	rec	f
wsj.sec24	97.04%	96.52%	97.02%	96.77%	97.17%	93.79%	95.45%
gweb-answers	89.43%	81.23%	87.85%	84.41%	88.93%	82.87%	85.79%
gweb-emial	89.14%	85.32%	78.96%	82.02%	88.67%	89.19%	88.93%
gweb-newsgroup	91.12%	85.87%	87.97%	86.92%	94.69%	88.42%	91.44%
gweb-reviews	91.76%	91.47%	88.90%	90.17%	92.44%	89.61%	91.00%
gweb-weblogs	94.46%	90.48%	93.16%	91.80%	92.91%	91.47%	92.19%

Table 2: Figures from full training file

2 Feature Design

```
featurestringlist.append(’FORM=%s’%token.form)
featurestringlist.append(’LEMMA=%s’%token.lemma)
featurestringlist.append(’SUFFIX=%s’%token.form[-3:])
featurestringlist.append(’CAPITALIZATION=%s’%token.form[0])
featurestringlist.append(’PLURAL=%s’%token.form[-1:])
featurestringlist.append(’DIGITWITHCH’ if token.form.isalnum == True else ’NOTDIGITWITHCH’)
featurestringlist.append(’DIGIT’ if token.form.isdigit == True else ’NOTDIGIT’)
featurestringlist.append(’CHARACTER’ if token.form.isalpha == True else ’NOTCHARACTER’)
```

The features I used include form, lemma, suffix (with length of 3) the first and last letter of a word, apart from that whether the word is number or alphabet or the combination of two which might indicate url is also checked. Here features all uppercase and word length are not included because they seems to decrease the accuracy for testing files in my case. And prefix with length longer than one also decrease the accuracy.

```
form1a = sentence[tid+1].form if tid < len(sentence)-1 else ’<FORM-END>’
form1b = sentence[tid-1].form if tid >0 else ’<FORM-BEGIN>’
form2a = sentence[tid+2].form if tid < len(sentence)-2 else ’<FORM-END>’
form2b = sentence[tid-2].form if tid >1 else ’<FORM-BEGIN>’
featurestringlist.append(’FORM1B=%s’%form1b)
featurestringlist.append(’FORM1A=%s’%form1a)
featurestringlist.append(’FORM2B=%s’%form2b)
featurestringlist.append(’FORM2A=%s’%form2a)
featurestringlist.append(’FORM1ASUFFIX=%s’%form1a[-3:])
featurestringlist.append(’FORM1BSUFFIX=%s’%form1b[-3:])
featurestringlist.append(’FORM1APREFIX=%s’%form1a[0])
featurestringlist.append(’FORM1BPREFIX=%s’%form1b[0])
```

For context information I checked the surrounding 1 and 2 words for the word in question. If the word is the first word the previous one word will be labelled as "FORM-BEGIN". If the word is the last word of the sentence, the following one will be labelled as "FORM-END". For larger context, for example with 3 surrounding words the accuracy will on the contrary be decreased. Further, I also try to mark the last 3 letters of a previous word and the following word so to distinguish the comparative and the superlative adjective phrases and gerund phrases. The reason to get the first letter of previous one word and the following one word is to detect "." as sentence end instead of abbreviation and "a" and "the" as determiner.