Natural Language Processing — Fall 2007 (Exercise 1)

1 Regular Expressions (5 + 5 = 10 Points)

(a) Solve exercises 2.1.d and 2.1.g from Jurafsky & Martin (2008).

(b) Solve exercise 2.8 from Jurafsky & Martin (2008).

2 Language Modelling (5 + 35 = 40 Points)

(a) Solve exercises 4.1 from Jurafsky & Martin (2008).

(b) Search the web for either the CMU SLM toolkit or the SRI Language Modelling toolkit. Download and compile a copy (best done on one of the IFI Linux servers) and familiarize yourself with the documentation. Personally, I found it more straightforward compiling a functional version of the CMU toolkit.

Next, obtain a plain-text copy of the Brown corpus from the following URL:

```
http://dingo.sbs.arizona.edu/~hammond/ling696f-sp03/browncorpus.txt
```

With these toys in place, determine the top-ten list of most frequent tokens in the Brown corpus. How many tokens occur exactly once? What are the total token and type counts for this corpus? Use a text processing tool like `sed`, `awk`, `emacs`, or `perl` (if you really have to) to improve the tokenization of our corpus. Aim to strip off, among other things, commas, sentence-final punctuation, and the possesive or contracted auxiliary `'s`. Without doing 'damage' to the corpus, how far can you bring down the type count, and how exactly did you do that?

Next, train a trigram language model from the complete Brown corpus. Use the LM toolkit to determine the perplexity for each of the following sentences (viewing one sentence at a time as a baby test 'corpus'):

```
That other steep path towards Bergen is short .
The other steep path towards Bergen is short .
The steep other road towards Bergen is short .
The second steep path towards Bergen is a card .
That other steep path against Bergen is short .
That steep other road against Bergen is short .
The steep second road towards Bergen is short .
The second steep road towards Bergen is a card .
The steep second path against Bergen is a card .
That steep second path against Bergen is short .
The other steep street towards Bergen is short .
That steep second street towards Bergen is short .
The steep second street towards Bergen is a card .
The second steep street towards Bergen is a card .
The steep second street towards Bergen is short .
That second steep street towards Bergen is short .
The second steep street towards Bergen is short .
```

All of the above are candidate translations, produced by the LOGON MT system for the input *Den andre bratte veien mot Bergen er kort.* Assuming that the goal of MT was to produce the most fluent output, how could we use the language model to rank candidate translations? Which of the above would seem the best output then? Does the picture change when you add context cues `<s>` and `</s>` to the training and test data?

3 Hidden Markov Models (7 + 7 + 6 = 20 Points)

Assume the following part-of-speech tagged training 'corpus'

hvis	jeg	hadde	hatt	min	gamle	hatt	,	så	hadde	jeg	hatt	hatt	.
CONJ	PRN	AUX	VB	POSS	ADJ	NN	DL	CONJ	AUX	PRN	VB	NN	DL

(a) In a few sentences, discuss the concept of smoothing and explain why it is important. Next, ignoring smoothing and making the standard simplifying assumptions for a bigram HMM, calculate the following:

(i) For each tag t, the probability of t following the tag VB, i.e. $P(t|\text{VB})$

(ii) For each word w, the probability of w given tags CONJ or VB, i.e. $P(w|\text{CONJ})$ and $P(w|\text{VB})$.

(b) Assuming further that our training corpus demonstrates our complete set of PoS tags, and further assuming that for each tag t $P(t|\texttt{<s>}) = P(t)$, construct part of the Viterbi trellis for tagging the sentence *jeg hadde hatt .* Rather than calculating all values, indicate the total size of the trellis and the computations for filling in the first two columns.

(c) In a few sentences, summarize the key points of the Viterbi algorithm. What is the interpretation of each cell in the trellis? What is the complexity of the algorithm, i.e. the number of computations performed in relation to (i) the length of the input sequence and (b) the size of the tag set? Discuss the naïve method of computing the most probable tag sequence t_1^n, given an input string w_1^n very briefly; state how the Viterbi algorithm improves over this approach.

4 Mirror English (6 + 14 = 20 Points)

Consider the language defined by the following grammar:

S → VP NP	NP → *kim*
VP → PP VP	NP → *oslo*
NP → PP NP	NP → *snow*
VP → NP V	V → *adores*
PP → P NP	P → *in*

(a) For each of the following items, identify the number of readings (distinct analyses) that the grammar of Mirror English assigns:

(i) *in oslo snow adores kim.*

(ii) *kim adores snow in oslo.*

(iii) *snow adores in oslo kim.*

(b) Where possible, provide one example each of a sentence of Mirror English with exactly (i) three, (ii) four, and (iii) five readings.

5 A Research Summary (10 Points)

Provide a brief summary of the research article that you have picked for presentation in class later this term. Use between half a page of text and at most one page.

Submit your results in email to Stephan by noon on Wednesday, October 17.