Prof. T. Hofmann, Dr. M. Ciaramita

Final Exam

August 25th, 2016

First and Last name: ______________________

Student ID (Legi) Nr: ______________________

Signature: ______________________

General Remarks

- Please check that you have all 20 pages of this exam.
- There are 74 points, and the exam is 120 minutes. **Don't spend too much time on a single question!** The maximum of points is not required for the best grade!
- Remove all material from your desk which is not permitted by the examination regulations.
- Write your answers directly on the exam sheets. If you need more space, make sure you put your **student-ID**-number on top of each supplementary sheet.
- Immediately inform an assistant in case you are not able to take the exam under regular conditions. Later complaints are not accepted.
- Attempts to cheat/defraud lead to immediate exclusion from the exam and can have judicial consequences.
- Please use a black or blue pen to answer the questions.
- Provide only one solution to each exercise. Cancel invalid solutions clearly.

	Topic	Max. Points	Points Achieved	Visum
1	Parsing	25		
2	Language Models	9		
3	HMMs	12		
4	Lexical Semantics	9		
4	Translation	8		
5	Neural Networks for NLP	11		
Total		74		

Grade: ....

1 Parsing

1.1 Deterministic Parsing

a) Why is the Chomsky Normal Form a requirement for the version of CKY you have seen?

1 pts

b) Name one important difference in the problem of constituency parsing and dependency parsing.

1 pts

c) After performing the CKY algorithm and filling out the table, how can one recognize that there are several valid parse trees that yield the given sentence?

1 pts

d) Contrast bottom-up and top-down parsing by explaining the ideas that are unique to the two approaches.

2 pts

e) Observe the following toy grammar:

6 pts

S $\rightarrow$ NP VBZ	DT $\rightarrow$ the
S $\rightarrow$ NP VP	NN $\rightarrow$ chef
VP $\rightarrow$ VP PP	NNS $\rightarrow$ fish
VP $\rightarrow$ VBZ NP	NNS $\rightarrow$ chopsticks
VP $\rightarrow$ VBZ PP	VBP $\rightarrow$ fish
VP $\rightarrow$ VBZ NNS	VBZ $\rightarrow$ eats
VP $\rightarrow$ VBZ NP	IN $\rightarrow$ with
VP $\rightarrow$ VBP NP	
VP $\rightarrow$ VBP PP	
NP $\rightarrow$ DT NN	
NP $\rightarrow$ DT NNS	
PP $\rightarrow$ IN NP	

Perform a CKY parse of the sentence *the chef eats fish with the chopsticks* by filling out the chart completely and drawing the final parse tree. Pseudo code of CKY can be found in the appendix.

the	chef	eats	fish	with	the	chopsticks

f) The sentence is an example of a class of syntactic ambiguity you have seen in the class. What class of ambiguity is it?

1 pts

g) Because of the very limited grammar this ambiguity is not present in the parse chart, yet. Which rule needs to be added to the grammar so that CKY yields the second tree as well?

2 pts

1.2 Probabilistic Parsing

a) Name one benefit of PCFGs over CFGs.

1 pts

b) Modify the CKY algorithm in the appendix to handle PCFGs in such a way that the most likely parse is computed. Give the full algorithm in pseudo code.

5 pts

c) How would the probabilistic CKY algorithm need to be modified so that it computes $P(s)$ where s is the input sentence. *(Just mention the modification, no pseudo code needed)*

2 pts

d) The following three replacement rules can be used to transform a CFG into Chomsky Normal Form:

- $A \rightarrow B$ is substituted with $A \rightarrow \beta_0 ... \beta_N$ for each B, $B \rightarrow \beta_0$ ... β_N (Unit rule replacement)
- $A \rightarrow \beta_0 ... \beta_i b \beta_j ... \beta_N$ is substituted with $A \rightarrow \beta_0 ... \beta_i B \beta_j ... \beta_N$ and $B \rightarrow b$ (Lexical replacement)
- $A \rightarrow \beta_0 ... \beta_{N-2} \beta_{N-1} \beta_N$ is substituted with $A \rightarrow \beta_0 ... \beta_{N-2} B$ and $B \rightarrow \beta_{N-1} \beta_N$

The three rules can be extended to handle PCFGs. Modify the rules to handle rule probabilities correctly.

3 pts

2 Language Models

2.1 Bi-gram model

Given a small training corpus

```
I am Sam
Sam I am
I do not like green eggs and ham
```

build a bi-gram language model to predict the probability of the following test sentence:

```
Sam does not like red apples
```

To deal with unseen bi-grams, apply add-one (Laplace) smoothing. Assume that the vocabulary contains all words in the training and test set. You do not need to compute probabilities that you do not need for the test sentence. You should carry out additions, but you do not have to do multiplications (e.g. something like $\frac{2}{3} \cdot \frac{1}{4} \cdot \frac{1}{3} \ldots$ is a valid answer)

4 pts

2.2 Perplexity

To evaluate language models, perplexity is the metric used most often. Answer the following questions (no negative points here).

1+1+1 pts

a) True or false: Perplexity is, numerically, the same as entropy.

b) True or false: Higher perplexity is better.

c) When the word dictionary size is 10,000, what perplexity does the random-guess approach give? Give an explanation or a short calculation.

2.3 Backoff

Backoff techniques for smoothing n-gram distributions fall back to lower order n-grams whenever counts are zero. Consider the 3-gram $w_1 w_2 w_3$ as an example. If $c(w_1, w_2, w_3) = 0$, backoff will try to use $P(w_3|w_2)$ instead of $P(w_3|w_2, w_1)$.

In addition to the "fall back" mechanism above, backoff techniques incorporate discounting. Why?

2 pts

3 HMMs

3.1 HMMs for PoS-tagging

a) Name a task that benefits from PoS tags. Justify your choice in one sentence.

1 pts ☐

b) Give an example for ambiguity in PoS tagging.

1 pts ☐

c) In HMM PoS tagging, we make two assumptions in addition to time invariance that allow for two approximations. Write down these two assumptions in words and in formulas.

2 pts ☐

d) What problem does the Viterbi algorithm solve?

1 pts ☐

e) The alpha quantity in the forward algorithm is defined as

$$\alpha_j^t = P(x^{1:t}, q^t = q_j)$$

Derive the formulation for α_j^t that allows using dynamic programming.

$\alpha_j^t =$

3 pts $\square$

3.2 HMMs for detecting PoS Sequences

We are interested in adjective(JJ)-noun(NN) sequences in a text. You are given an HMM for PoS tagging parametrized by state transition probabilities a_{ij} and emission probabilities $b_i(o)$. Say state i corresponds to JJ and state j corresponds to NP. Given a sentence $O = o_1, \ldots o_n$, what is the probability of observing an adjective-noun sequence in this sentence? You can reuse quantities you have seen in the lecture but explain very briefly what they correspond to.

4 pts $\square$

4 Lexical Semantics

4.1 word2vec & GloVe

Which of the following claims are true/false? *(1 point per correct answer, -1 point per incorrect answer, 0 per blank, non-negative total points in any case)*

5 pts

a) Glove has significantly higher memory requirements than word2vec.
[ ] True [ ] False

b) Glove's weight function upweights small counts that typically carry fine-grained semantic information.
[ ] True [ ] False

c) The logarithm in GloVe's cost function makes optimization significantly harder.
[ ] True [ ] False

d) In contrast to GloVe, word2vec's cost function cannot be optimized using SGD.
[ ] True [ ] False

e) Antonyms (e.g. cheap and expensive) are often embedded closely to each other.
[ ] True [ ] False

4.2 Sentence Embeddings

a) In the lecture you have seen co-occurrence-based methods for lexical semantics such as GloVe. Suppose we want to use the same idea to embed sentences of a fixed length, say 10. Analogous to words, we can define co-occurrences of such sentences. Name exactly two problems with this approach.

2 pts

b) The word2vec cost function with negative sampling is given as:

$$\mathcal{L}(\theta) = \sum_{(i,j)\in\triangle^+} \log\sigma(\langle \mathbf{x}_i, \mathbf{y}_j\rangle) + \sum_{(i,j)\in\triangle^-} \log\sigma(-\langle \mathbf{x}_i, \mathbf{y}_j\rangle)$$

Name a problem with the objective that arises when the negative samples are removed.

1 pts

c) We want to use GloVe to learn word vectors on a corpus with n distinct words. Which empirical property of language can help us to estimate the number of non-zero entries in the co-occurrence matrix.

1 pts $\boxed{}$

5 Translation

5.1 A Simple Probabilistic Model

Let's consider a simple model for translating English into French. Let $f_i \in V_F$ denote French words and $e_j \in V_E$ denote English words. Furthermore, given an alignment a and a French word f_i let's write e_{a_i} as the English word that f_i is aligned to via a.

The model builds on the common simplifying assumption that the French words are independent given their aligned English source word. The translation probabilities are parametrized directly by parameters t as

$$P(f_i|e_j) = t(f_i|e_j) \in \mathbb{R} \quad \forall i \in V_F, j \in V_E$$

Throughout the exercise, use E, F and A as the random variables realizing English sentences, French sentences and sentence alignments. For simplicity we do not give the details of how the alignment model is parametrized, so you can use $P(A|E)$ at any point.

a) Using the assumptions and parametrizations from above, spell out the following probability in terms of the model probabilities:

2 pts ☐

$P(A, F|E) =$

b) Using the probability above, write out the actual translation probability. *(You can solve this even though you did not solve part a)*

1 pts ☐

$P(F|E) =$

c) What is the latent variable in this model?

1 pts ☐

5.2 Expectation-Maximization for Learning

If we are given a word-aligned corpus, we can compute counts $\mathrm{count}(f_i|e_j)$ from the corpus to estimate parameters $t(f_i|e_j)$. However, if we only have a sentence-aligned corpus, we need to resort to an Expectation-Maximization algorithm. Here we develop the E-step.

First, state precisely what expectation we want to compute.

Second, write out the expectation using only the two quantities from 5.1 a) and b)

(You can solve this even though you did not solve parts a) and b) of 5.1)

4 pts

6 Neural Networks for NLP

6.1 Neural Language Models

a) Neural language models (NLMs) address the sparsity problem of n-grams in a fundamentally different way (compared to n-gram models). What is the key idea?

2 pts

b) The idea of NLMs has been extended from the word-level to the character-level. That means, we predict the next character instead of the next word. Besides some quantitative aspects (memory, performance, etc.) this allows to solve a fundamental problem that traditional models can't address easily. Which one?

1 pts

6.2 Convolutional Neural Networks

a) Convolutional neural networks (CNNs) are very popular networks to derive fixed-length sentence vectors. Explain the significance of the filter width when applied to text.

1 pts

b) Running a convolution with a filter of width 3 over a sentence of length $n > 3$ yields $n - 2$ values (since we do not do padding here). How does the typical CNN architecture manage to provide a length-independent sentence vector.

1 pts

6.3 Recursive Neural Networks

Recursive neural networks (e.g. for sentiment classification) represent words as vectors and make the composition function, which turns the meaning of two parts into one, explicit.

For word vectors $x, y \in \mathbb{R}^d$ we propose three composition functions $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$

$$f_1(x, y) = x + y$$

$$f_2(x, y) = W_1 x + W_2 y \qquad \text{where } W_1, W_2 \in \mathbb{R}^{d \times d}$$

$$f_3(x, y) = x\mathbf{W}y \qquad \text{where } \mathbf{W} \in \mathbb{R}^{d \times d \times d}$$

1+1+1 pts ☐

a) Name an advantage of f_2 over f_1

b) Name an advantage of f_3 over f_2.

c) Name a disadvantage of f_3 compared to f_1.

6.4 Neural Document Representations

In the class you have seen the distributed memory vectors approach to neural document modelling. This approach employs a standard language modelling architecture that predicts a word given its context's word vectors:

$$p(w_{j_t} | d, w_{j_{t-1}} \cdots w_{j_{t-L}}) = \frac{\exp\left[\langle \mathbf{z}_{j_t}, \mathbf{c} \rangle\rangle\right]}{\sum_{j=1}^{M} \exp\left[\langle \mathbf{z}_j, \mathbf{c} \rangle\rangle\right]}$$

where $\mathbf{z}$ are the soft-max weights. In contrast to plain language models, the context vector $\mathbf{c}$ is the concatenation of the context vectors *and* a document vector $\mathbf{c} = (\mathbf{y}_d^\top, \mathbf{x}_{j_{t-1}}^\top, \ldots, \mathbf{x}_{j_{t-L}}^\top)^\top$. Answer the following questions (no negative points here).

1+1+1 pts

a) True or false: The purpose of the document vector is to learn features that are semantically orthogonal to the words in the context window.

b) True or false: When making a prediction for a new document from the test set, the model needs to re-train on that document.

c) True or false: The document vector can have a different dimensionality than the word vectors $\mathbf{x}_i$.

A Algorithms

CKY

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  $table[j-1, j] \leftarrow \{A \mid A \rightarrow words[j] \in grammar\}$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      $table[i,j] \leftarrow table[i,j] \cup$
        $\{A \mid A \rightarrow BC \in grammar,$
          $B \in table[i,k],$
          $C \in table[k,j]\}$

Algorithm 1: CKY Parsing for CFGs in CNF

Supplementary Sheet

Supplementary Sheet

Supplementary Sheet