UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR11062 MACHINE TRANSLATION (LEVEL 11)

Wednesday 11 th May 2016

09:30 to 11:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 4 Courses

Convener: I. Stark
External Examiners: A. Burns, A. Cohn, P. Healey, T. Field, T. Norman

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. Probabilistic Word Alignment

(a) Given a foreign sentence $f = f_1...f_I$ and English sentence $e = e_1...e_J$, a probabilistic translation model must define the conditional probability $p(f|e)$. Using the chain rule, we can write this as:

$$p(f|e) = p(I|J,e)\prod_{i=1}^{I} p(f_i|f_{i-1},...,f_1,e,I,J) \tag{1}$$

At this point, in classical (i.e. information theoretic) probabilistic models like the IBM Models, we typically introduce a latent variable $a = a_1...a_I$, where each a_i is a random variable taking a value in $1,...,J$ defining the *alignment* of the ith French word (note that there are no null alignments). We then obtain a new model of f and a conditioned on e

$$p(f,a|e) = p(I|J,e)\prod_{i=1}^{I} p(a_i|a_{i-1}...a_1,f_{i-1}...f_1,e,I,J)\cdot p(f_i|a_i...a_1,f_{i-1}...f_1,e,I,J) \tag{2}$$

What step do we now take to obtain a model like IBM Model 1, and how does the introduction of the latent alignment variable allow us to do this? In other words, what technical problem in the parameterisation of Equation 2 does the latent alignment variable solve? [*2 marks*]

(b) Recurrent neural translation models do not include any latent alignment variables. How do they work around the problem that the latent variables are meant to solve? [*2 marks*]

(c) Given the model in Equation 2, how many different assignments are there to the latent variable a for a French sentence with I words and an English sentence of J words? Explain your reasoning. [*2 marks*]

(d) The definition of the latent variable a treats English and French words differently, and some alignments cannot be represented by any assignment to a. Describe these alignments and explain why a cannot represent them. [*3 marks*]

QUESTION CONTINUES ON NEXT PAGE

(e) Suppose that we train IBM Model 1 using the Expectation Maximization (EM) algorithm. The model crucially depends on translation parameters $p(f_i|e_j)$ learned from data, where f_i is a French word and e_j an English word. Now consider the English words *house* and *houses.* What is the relationship between the translation distributions $p(\cdot|\mathit{house})$ and $p(\cdot|\mathit{houses})$? Is any relationship (or lack of one) due to properties of the model, learning algorithm, or data? [*3 marks*]

(f) Suppose we train IBM Model 1 using Expectation Maximization (EM) on data in which the word *house* appears 100 times and the word *larch* appears once. Sketch their translation distributions, explain how the model and learning algorithm interact to produce the distributions, and explain how the distributions affect the alignments of these words. Describe a strategy to counteract the effect on alignments and explain why you think it will work. [*5 marks*]

(g) The Nunavut legislative assembly in Canada publishes many parallel documents in English and Inuktitut, a native North American language. The IBM Models don't work very well on this data because Iniktitut is highly agglutinative. For example, the word *qangatasuukkuvimmuuriaqalaaqtunga* is composed of a succession of distinct morphemes and is roughly equivalent to *I'll have to go to the airport.* This breaks the word-to-word alignment assumption of the IBM Models, so we need to align morphemes to words. Assuming that you have no morphological analyzer, morphological dictionary, or additional data available to you, explain how you would modify IBM Model 1 to align English words to subwords of the Inuktitut representing probable morphemes. You can model Inuktitut conditioned on English or the other way around (though you'll probably find the latter direction simpler). What latent variables would you introduce and how would you model them? Write a precise mathematical description of your model, and explain how you would apply it to the data to learn its parameters using EM. What difficulties might arise in doing this? [*8 marks*]

2. Decoding via Dynamic Programming

(a) We have learned a lexical translation model in which each German word in the left column can translate to any of the English words in the right column of the corresponding row.

auch	either | also | too
aufgerufen	call | called | invoked
haben	did | have | hold
mich	me | myself | I
nicht	not | no | non
sie	you | she | her

Now we see this German input sentence:

sie haben mich auch nicht aufgerufen

Each word of the input sentence must be translated exactly once. How many translations are there if we translate words in order from left to right? You can write an expression that produces the answer. You do not need to calculate the result. *[3 marks]*

(b) Enumerating all of these translations to find the best is inefficient, so we will use dynamic programming. Assume our model has two features:

- Feature w_{LM} decomposes over English bigrams, so that for output sentence $e = e_1...e_J$:

$$w_{LM}(e) = w_{LM}(e_1, \langle\text{start}\rangle) + \left(\sum_{j=2}^{J} w_{LM}(e_j, e_{j-1})\right) + w_{LM}(\langle\text{stop}\rangle, e_J)$$

- Feature w_{TM} decomposes over aligned word pairs. For foreign sentence $f = f_1...f_I$ and English sentence $e = e_1...e_J$ define alignment $a = a_1...a_I$, with each $a_i \in 1, ..., J$ such that:

$$w_{TM}(e, a, f) = \sum_{i=1}^{I} w_{TM}(e_{a_i}, f_i)$$

To simplify matters, assume each English word aligns to exactly one foreign word (hence $I = J$). Let $E(f)$ be the set of translations of word f, e.g. $E(\text{auch}) = \{\text{either, also, too}\}$. Our dynamic program reasons over items of the form $h[i, e']$, where $i \in 1, ..., I$ is a position in the source sentence and e' is the most recently produced English word. Item Goal represents a complete translation. Compute weight W of the best translation recursively as:

QUESTION CONTINUES ON NEXT PAGE

$$
\begin{aligned}
W(h[0, \langle \text{start} \rangle]) =& 0 \\
\forall_{i \in 1,\ldots,I} \forall_{e' \in E(f_i)} W(h[i, e']) =& \max_{e'' \in E(f_{i-1})} W(h[i-1, e'']) + w_{LM}(e', e'') + w_{TM}(e', f_i) \\
W(\textsc{Goal}) =& \max_{e' \in E(f_I)} W(h[I, e']) + w_{LM}(\langle \text{stop} \rangle, e')
\end{aligned}
$$

What is the complexity of this dynamic program? Explain your reasoning. [*3 marks*]

(c) How would you change the dynamic program if we used a trigram language model instead of a bigram language model, and how would your change affect decoding complexity? [*3 marks*]

(d) How many translations are there if we can translate words in any order? Explain your reasoning. [*2 marks*]

(e) Translating words in order is too strict, but arbitrary permutation is too permissive. Let's explore a middle ground, the **IBM constraint**: we only translate a word if there are d or fewer untranslated words to its left. For example, if $d = 1$, we only translate *auch* if at least two of *sie*, *haben*, and *mich* have been translated. Write a dynamic program for $d = 1$. (Ask yourself: What possible items can the dynamic program have? For each item, what are all possible ways of reaching it in one step?) [*6 marks*]

(f) Even with dynamic programming, translation may be too slow for large enough values of d. One way to solve this is with pruning low-probability states in an equivalence class. Define a reasonable equivalence class for your dynamic program and explain your reasoning. [*4 marks*]

(g) What is a tradeoff of pruning? [*2 marks*]

(h) Recurrent neural translation models do not use dynamic programming. Why not? [*2 marks*]

3. Language, Loss Functions, and Features

(a) Suppose that we have the following input sentence:

Er hätte der Berichterstattung gegenber dem Parlament Vorrang einräumen müssen .

A human translator has provided us with the following reference sentence:

He should have prioritised giving that report to Parliament .

A machine translation system produces the following output:

He should have the reporting to the Parliament to give priority .

The **word error rate** (WER), **edit distance**, or **Levenstein distance** is a common measure of string similarity. Passing left-to-right through a string pair, it asks for the minimum number of word insertions, deletions, or substitutions that can be used to produce the second string from the first. It is used to measure accuracy for automatic speech recognition and optical character recogntion, which, like machine translation, produce strings as output. One view of WER is that it seeks an alignment of the string pair that minimizes the number of aligned tokens that differ from each other, under the constraint that alignment links do not cross each other.

What is the word error rate of the machine translation output when compared to the human translation in the above example? Draw the corresponding alignment. [*2 marks*]

(b) What are pros and cons of word error rate? Illustrate at least one of them using the example. [*2 marks*]

(c) BLEU relaxes word error rate to account for the fact that good translations can come with many different word orders, but the computation of BLEU still relies on exact word match. Exact word match is a reasonable assumption for English, but less reasonable for other languages. Describe three different linguistic phenomena prominent in other languages that interact poorly with the assumption of exact word match. Explain your reasoning. (It is ok if the phenomena you describe are all at the same level, e.g. all morphological phenomena, or all syntactic phenomena, provided that you make it clear how they are distinct).

[*6 marks*]

QUESTION CONTINUES ON NEXT PAGE

(d) Most translation systems are defined as linear models over an English sentence e, French sentence f, and alignment a. The central statistical model is $s(e, a|f) = \theta \cdot h(e, a, f)$, where θ is a parameter vector, $h(e, a, f)$ is a fixed feature vector over input-output pairs, and the resulting $s(e, a|f)$ is the dot product of the parameter and feature vectors. A frequent feature in these systems is the output length $|e|$. How does this length feature interact with BLEU? *[2 marks]*

(e) *Agreement* is a widespread phenomenon in which a word is inflected to reflect some property of another word with which it is in a (usually grammatical) relationship. For example, verbs are often inflected to agree with one of their arguments (in English, the subject):

i. He/she/it jumps.

ii. I/we/you/they jump.

English has quite simple inflectional morphology; in some languages, each different subject of the above example would require a different verb form. For a machine translation system to produce fluent output in these languages, it must handle agreement. In which cases would agreement be handled correctly in a phrase-based machine translation system? In which cases would it fail? *[3 marks]*

(f) We would like to encourage agreeement in the output language of a machine translation system. How could you incorporate an explicit model of agreement into one of the models that we discussed in class? Identify the parameters of this model, and explain how you would learn them. Explain how the new aspects of your model would interact with decoding. Would you need to modify the decoding algorithm? Explain your reasoning. *[6 marks]*

(g) Devise an experiment to test whether your agreement model is successful, and explain what experiments you would run. What would you measure? Explain your reasoning. *[4 marks]*