Master on Artificial Intelligence Universitat Politècnica de Catalunya

Advanced Human Language Technologies

Final Exam June 2^{nd}, 2022

Exercise 1. Estimation & Smoothing

Worldwide online retail seller Papazom.com asked us to design a model that classifies products into a set of categories, given the product description. The target categories are: *Electronics* (ELEC), *Computers* (COMP), *Fashion* (FASH), and *Tools* (TOOL).

We have a sample of 1,500 annotated product descriptions. 200 are classified as ELEC, 350 as COMP, 650 as FASH, and 300 as TOOL.

- 70 products contain the word *display*. 40 are annotated as ELEC, 20 as COMP, and 10 as FASH.
- 150 products contain the word *Gigabytes*. 90 are annotated as ELEC, and 60 as COMP.
- 160 products contain the word *waterproof*. 80 are annotated as FASH, 30 as ELEC, and 50 as TOOL.
- 110 products contain the word *handmade*. 95 are annotated as FASH, and 15 as TOOL.

1. Compute the following probability values corresponding to a MLE model
   - Probability that a product belongs to category *Electronics*, $P_{MLE}(\text{ELEC})$
   - Probability that a product containing word *handmade* is in *Fashion*, $P_{MLE}(\text{FASH}|handmade)$
   - Probability that a product containing word *handmade* is in *Computers*, $P_{MLE}(\text{COMP}|handmade)$
   - Probability that a *Computer* product contains word *display*, $P_{MLE}(display|\text{COMP})$
   - Probability that a *Computer* product contains word *handmade*, $P_{MLE}(handmade|\text{COMP})$
2. Compute the following probability values corresponding to a model smoothed using absolute discount with $\delta = 0.3$
   - Probability that a product description contains word *waterproof*, $P_{AD}(waterproof)$
   - Probability that a product is in *Tools* and contains word *handmade*, $P_{AD}(\text{TOOL} \wedge handmade)$
   - Probability that a product containing word *display* is in *Electronics*, $P_{AD}(\text{ELEC}|display)$
   - Probability that a product containing word *display* is in *Tools*, $P_{AD}(\text{TOOL}|display)$
3. Compute the following probability values corresponding to a model smoothed with Lidstone's Law with $\lambda = 0.2$.
   - Probability that a product belongs to category *Tools*, $P_{LID}(\text{TOOL})$
   - Probability that a product containing word *Gigabytes* is in *electronics*, $P_{LID}(\text{ELEC}|Gigabytes)$
   - Probability that a product containing word *Gigabytes* is in *Tools*, $P_{LID}(\text{TOOL}|Gigabytes)$

Justify your answer and the values chosen for B, N, and N_0 where it applies.

Develop your computations, do not provide just a numeric result. You may leave probabilities as fractions.

SOLUTION

1. Maximum Likelihood

$$P_{MLE}(\text{ELEC}) = \frac{\#\text{ELEC}}{N} = \frac{200}{1500} \quad (1)$$
$$P_{MLE}(\text{FASH}|handmade) = \frac{\#(\text{FASH} \wedge handmade)}{\#handmade} = \frac{95}{110} \quad (2)$$
$$P_{MLE}(\text{COMP}|handmade) = \frac{\#(\text{COMP} \wedge handmade)}{\#handmade} = \frac{0}{110} \quad (2)$$
$$P_{MLE}(display|\text{COMP}) = \frac{\#(\text{COMP} \wedge display)}{\#COMP} = \frac{20}{350} \quad (3)$$
$$P_{MLE}(handmade|\text{COMP}) = \frac{\#(\text{COMP} \wedge handmade)}{\#COMP} = \frac{0}{350} \quad (3)$$

2. Absolute Discount

$$P_{AD}(waterproof) = \frac{\#waterproof - \delta}{N} = \frac{160 - 0.3}{1500} \quad (1)$$
$$P_{AD}(\text{TOOL} \wedge handmade) = \frac{\#(\text{TOOL} \wedge handmade) - \delta}{N} = \frac{15 - 0.3}{1500} \quad (1)$$
$$P_{AD}(\text{ELEC}|display) = \frac{\#(\text{ELEC} \wedge display) - \delta}{\#display} = \frac{40 - 0.3}{70} \quad (4)$$
$$P_{AD}(\text{TOOL}|display) = \frac{(B - N_0)\delta/N_0}{N} = \frac{(4-1)0.3/1}{70} \quad (4)(5)(6)$$

3. Lidstone's Law

$$P_{LID}(\text{TOOL}) = \frac{\#ELEC + \lambda}{N + B\lambda} = \frac{300 + 0.2}{1500 + 4 \times 0.2} \quad (1)(5)$$
$$P_{LID}(\text{ELEC}|Gibabytes) = \frac{\#(ELEC \wedge Gigabytes) + \lambda}{N + B\lambda} = \frac{90 + 0.2}{150 + 4 \times 0.2} \quad (5)(7)$$
$$P_{LID}(\text{TOOL}|Gibabytes) = \frac{\#(TOOL \wedge Gigabytes) + \lambda}{N + B\lambda} = \frac{0 + 0.2}{150 + 4 \times 0.2} \quad (5)(7)$$

Justification of chosen values:

(1) $N = 1500$ because we have 1500 products, i.e. 1500 cases where a word or a class may be observed.

(2) We are conditioning on the occurrece of *handmade*, which occurrs in 110 products, so that is our number of observations N.

(3) We are conditioning on the occurrece of COMP, which occurrs in 350 products, so that is our number of observations N.

(4) We are conditioning on the occurrece of *display*, which occurrs in 70 products, so that is our number of observations N.

(5) We are computing the probability of the class, so there are 4 possible outcomes, thus $B = 4$.

(6) Only 3 classes (out of 4 possible) were observed with *display*, thus there is one unobserved class: $N_0 = 1$.

(7) We are conditioning on the occurrece of *Gigabytes*, which occurrs in 150 products, so that is our number of observations N.

Exercise 2. Distances/Similarities

Papazom.com also needs to match offers from different suppliers that correspond to the same product, as well as to match user queries with product descriptions.

For this, they asked us to propose a similarity model able to establish how similar two product description are.

For instance, given the product descriptions.

s_1	*smartphone Hoewai x23-A with latest super AMOLED display and 64Gb*
s_2	*smartphone x23-A with 64Gb and AMOLED charge indicator*
s_3	*Hoewai smartphone z21-B with super AMOLED display and 32Gb*

1. Represent each description as a word set, and compute $sim_{jac}(s_1, s_2)$, $sim_{jac}(s_1, s_3)$, and $sim_{jac}(s_2, s_3)$ using Jaccard similarity
2. Represent each description as a word-bigram set (i.e set elements are not single words, but word-bigrams in the sentece), and compute $sim_{cos}(s_1, s_2)$, $sim_{cos}(s_1, s_3)$, and $sim_{cos}(s_2, s_3)$ using Cosine similarity.
3. A Papazon.com user wrote the search *Hoewai smartphone AMOLED display*. Compute the similarities of this query with s_1, s_2, and s_3 with each of the above metrics (unigram Jaccard and bigram Cosine).

SOLUTION

1.

	s_1	s_2	s_3
smartphone	1	1	1
Hoewai	1	0	1
x23-A	1	1	0
with	1	1	1
latest	1	0	0
super	1	0	1
AMOLED	1	1	1
display	1	0	1
and	1	1	1
64Gb	1	1	0
charge	0	1	0
indicator	0	1	0
z21-B	0	0	1
32Gb	0	0	1

$$sim_{jac}(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|} = \frac{6}{12} = 0.50$$

$$sim_{jac}(s_1, s_3) = \frac{|s_1 \cap s_3|}{|s_1 \cup s_3|} = \frac{7}{12} = 0.58$$

$$sim_{jac}(s_2, s_3) = \frac{|s_2 \cap s_3|}{|s_2 \cup s_3|} = \frac{4}{13} = 0.31$$

2.

	s_1	s_2	s_3
smartphone Hoewai	1	0	0
Hoewai x23-A	1	0	0
x23-A with	1	1	0
with latest	1	0	0
latest super	1	0	0
super AMOLED	1	0	1
AMOLED display	1	0	1
display and	1	0	1
and 64Gb	1	0	0
smartphone x23-A	0	1	0
with 64Gb	0	1	0
64Gb and	0	1	0
and AMOLED	0	1	0
AMOLED charge	0	1	0
charge indicator	0	1	0
Hoewai smartphone	0	0	1
smartphone z21-B	0	0	1
z21-B with	0	0	1
with super	0	0	1
and 32Gb	0	0	1

$$sim_{cos}(s_1, s_2) = \frac{|s_1 \cap s_2|}{\sqrt{|s_1|}\sqrt{|s_2|}} = \frac{1}{\sqrt{9}\sqrt{7}} = 0.13$$

$$sim_{cos}(s_1, s_3) = \frac{|s_1 \cap s_3|}{\sqrt{|s_1|}\sqrt{|s_3|}} = \frac{3}{\sqrt{9}\sqrt{8}} = 0.35$$

$$sim_{cos}(s_2, s_3) = \frac{|s_2 \cap s_3|}{\sqrt{|s_2|}\sqrt{|s_3|}} = \frac{0}{\sqrt{7}\sqrt{8}} = 0.00$$

3.

	query (q)
smartphone	1
Hoewai	1
x23-A	0
with	0
latest	0
super	0
AMOLED	1
display	1
and	0
64Gb	0
charge	0
indicator	0
z21-B	0
32Gb	0

$$sim_{jac}(s_1, q) = \frac{|s_1 \cap q|}{|s_1 \cup q|} = \frac{4}{10} = 0.40$$

$$sim_{jac}(s_2, q) = \frac{|s_2 \cap q|}{|s_2 \cup q|} = \frac{2}{10} = 0.20$$

$$sim_{jac}(s_3, q) = \frac{|s_3 \cap q|}{|s_3 \cup q|} = \frac{4}{9} = 0.44$$

	query (q)
smartphone Hoewai	0
Hoewai x23-A	0
x23-A with	0
with latest	0
latest super	0
super AMOLED	0
AMOLED display	1
display and	0
and 64Gb	0
smartphone x23-A	0
with 64Gb	0
64Gb and	0
and AMOLED	0
AMOLED charge	0
charge indicator	0
Hoewai smartphone	1
smartphone z21-B	0
z21-B with	0
with super	0
and 32Gb	0
smartphone AMOLED	**1**

$$sim_{cos}(s_1, q) = \frac{|s_1 \cap q|}{\sqrt{|s_1|}\sqrt{|q|}} = \frac{1}{\sqrt{9}\sqrt{3}} = 0.19$$

$$sim_{cos}(s_2, q) = \frac{|s_2 \cap q|}{\sqrt{|s_2|}\sqrt{|q|}} = \frac{0}{\sqrt{7}\sqrt{3}} = 0.00$$

$$sim_{cos}(s_3, q) = \frac{|s_3 \cap q|}{\sqrt{|s_3|}\sqrt{|q|}} = \frac{2}{\sqrt{8}\sqrt{3}} = 0.41$$

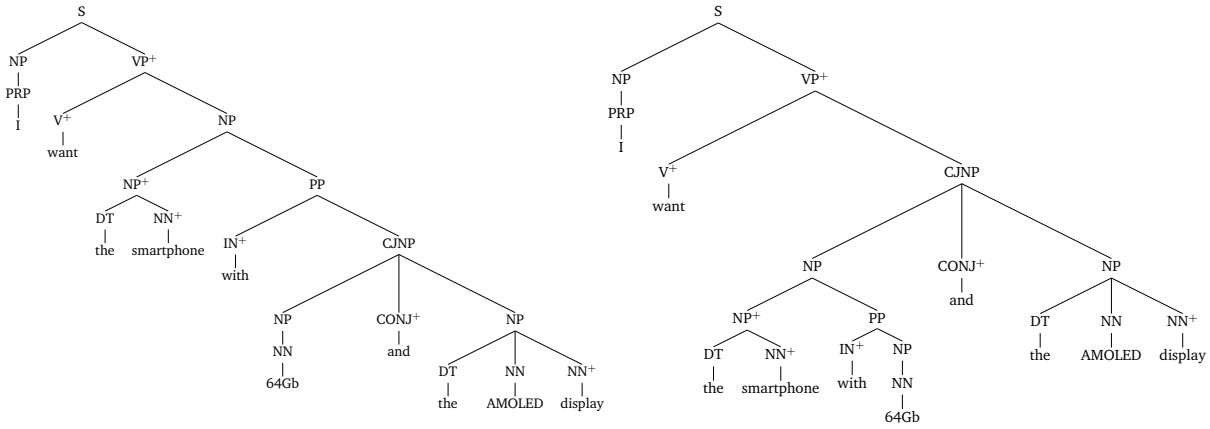
Exercise 3. Parsing

A Papazom.com user wrote the sentence:

I want the smartphone with 64Gb and the AMOLED display

We used the following PCFG grammar (where the $^+$ superscript indicates the head of each rule), and obtained the two possible parse trees below.

r_1 S $\rightarrow$ NP VP$^+$	1.0	r_7 NP $\rightarrow$ PRP	0.1	r_{12} PRP $\rightarrow I$	1.0
r_2 VP $\rightarrow$ V$^+$ NP	0.7	r_8 NP $\rightarrow$ NN	0.2	r_{13} V $\rightarrow$ *want*	1.0
r_3 VP $\rightarrow$ V$^+$ CJNP	0.3	r_9 NP $\rightarrow$ NP$^+$ PP	0.2	r_{14} CONJ $\rightarrow$ *and*	1.0
r_4 CJNP $\rightarrow$ NP CONJ$^+$ NP	1.0	r_{10} NP $\rightarrow$ DT NN$^+$	0.3	r_{15} DT $\rightarrow$ *the*	1.0
r_5 PP $\rightarrow$ IN$^+$ NP	0.6	r_{11} NP $\rightarrow$ DT NN NN$^+$	0.2	r_{16} IN $\rightarrow$ *with*	1.0
r_6 PP $\rightarrow$ IN$^+$ CJNP	0.4			r_{17} NN $\rightarrow$ *display*	0.3
				r_{18} NN $\rightarrow$ *smartphone*	0.4
				r_{19} NN $\rightarrow$ 64*Gb*	0.2
				r_{20} NN $\rightarrow$ *AMOLED*	0.1



1. Which parse tree has higher probability according to the PCFG? Reason your answer.
2. Convert both parse trees to dependency trees.
3. Describe in an unambiguous form what is the meaning of the interpretation represented by each tree.

SOLUTION

1. The probability of a parse tree is computed as $P(t) = \prod_{r \in t} q(r)$, that is, the product of the probabilities of the rules used to build it.

   Both trees use *each rule* in the grammar exactly *once*, with only two exceptions:

   - The first tree does not use r_3 nor r_5.
   - The second tree does not use r_2 nor r_6.

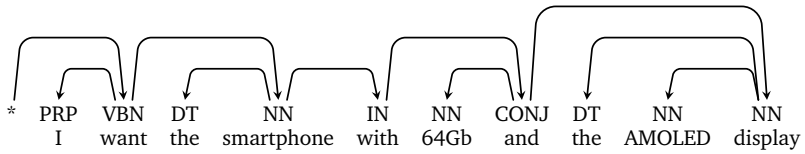
   Let Q be the product of the probabilities of the rules common to both trees,
   i.e. $Q = \prod_{r \in R - \{r_2, r_3, r_5, r_6\}} P(r)$.

   Then, the first tree will have probability $P_1 = Q \times P(r_2) \times P(r_6) = Q \times 0.7 \times 0.4 = Q \times 0.28$, while the second tree will have probability $P_2 = Q \times P(r_3) \times P(r_5) = Q \times 0.3 \times 0.6 = Q \times 0.18$.

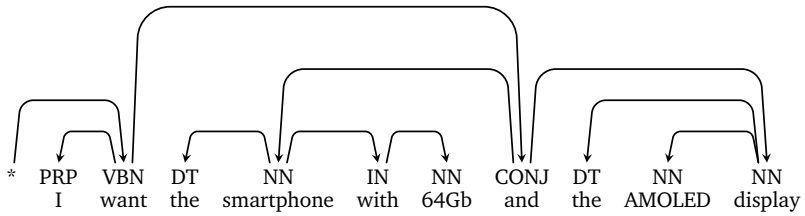
   Since $Q \times 0.28 > Q \times 0.18$, the first tree has higher probability according to this PCFG.

2. Convert both parse trees to dependency trees.

   First tree:



Second tree:



3. Describe in an unambiguous form what is the meaning of the interpretation represented by each tree.

First tree has the interpretation that the user wants a smartphone that has two features: 64Gb and an AMOLED display

The second tree interpretation is that the user wants two products: One smartphone with 64Gb, and also one AMOLED display.

Exercise 4. Word Embeddings

1. Describe the main differences between Word2Vec and GloVe.
2. The skip-gram model (Word2Vec) uses some tricks to speed up the training time. One such trick is *negative sampling*. Briefly describe how *negative sampling* helps reduce training time in the skip-gram model. Assume that the word-vectors' dimensionality is $d = 300$ and the vocabulary size is $|V| = 2 \cdot 10^6$.

SOLUTION

1. They differ in the way they are trained. GloVe is based on global word to word co-occurrence counts (in the entire corpus). Word2Vec uses co-occurrence within local context (neighbour words).

   More especifically, GloVe's training objective is to find a feature matrix that factorizes the whole word to word co-ocurrence matrix while keeping most of its variance. Word2Vec is trained using either the skip-gram model, wich tries to predict the context words given a central word; or the cbow model, which tries to predict the central word given all words in its context.

   Another aspect to consider is the fact that *GloVe*'s training time scales with the vocabulary size whereas *Word2Vec* scales with the corpus size.

2. Given a window of words from the corpus and a target word (central word), the skip-gram model's aim is to predict the context from that target word. The model typically inverts the contexts and targets, and tries to predict each context word from its target word, that is, the training proces uses pairs of (target_word, context_word$_i$) for every context word.

   Without the "negative sampling" trick, all word weights should be updated at each training step (the final softmax layer covers all words in the vocabulary so that every weight is affected by back-propagation). With "negative sampling", a subset of randomly selected words are used as negative examples and the model is only updated for the context word plus these negative examples.

   Without negative sampling, all $|V| \times d$ weights would be updated. With negative sampling, only $(n + 1) \times d$ weights would be updated, where n is the number of negative samples.

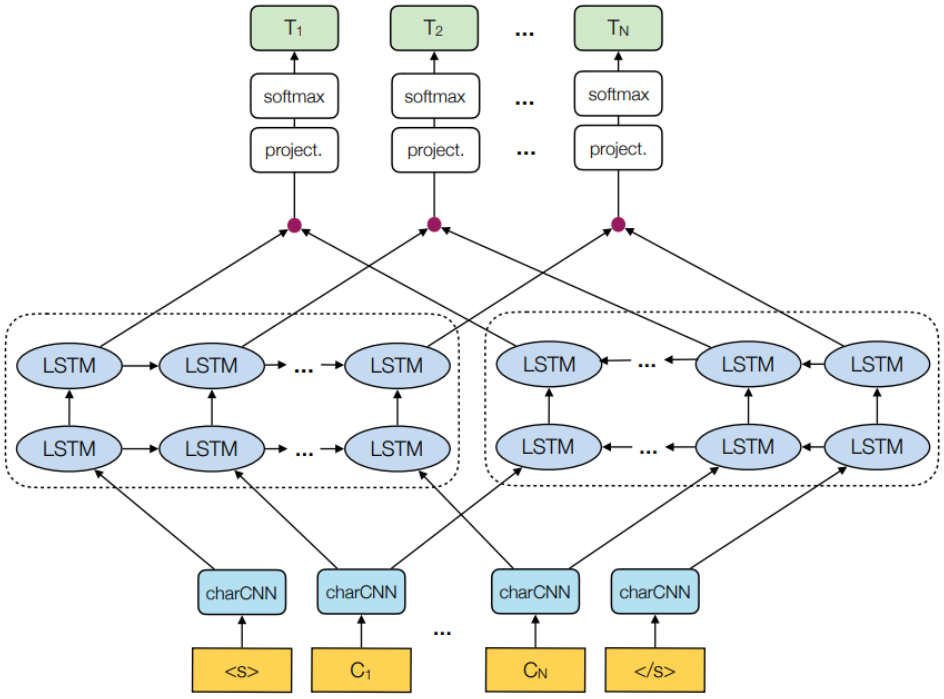
Exercise 5. Neural Networks

ELMo is a NLP model that uses character embeddings and two stacked bidirectional LSTM layers.

1. Draw the ELMo architecture for language modeling and compare it to a traditional one-layer RNN-based language model architecture (using word2vec word embeddings).
2. What are the advantages and disadvantages of the aforementioned RNN-based language model respect to ELMo?
3. The Hard Tangent and Rectified Linear Unit activation functions can help speed up training time. Why? Can you mention any drawback? How could these drawbacks be prevented?

SOLUTION

1.



(remember the projection and softmax layers)

Compared to a traditional RNN LM, ELMo uses memory units and is a bidirectional (not causal) language model. Another difference in this particular case is the use of pretrained word embeddings rather than word representations from CNN over character embeddings. * Causal RNN can be used for language generation.

2. Advantages:

   - Simpler (less weights)
   - Causal (Useful for language generation, easier to train)
   - Does not need pre-training for downstream tasks (uses pretrained word embeddings)

   Disadvantages:

   - Prone to vanishing gradients

- Causal (Ignores future words)
- No lexical information (lacks character embeddings)
- Less powerful for capturing higher-level abstractions (Just one layer)

3. Advantages:
   - Prevents vanishing gradients (no small values for the derivate, either 0 or m)
   - More computationally efficient (compared to Sigmoid-like functions) since Relu just needs to pick $max(0, x)$ and not perform expensive exponential operations as in Sigmoids.
   - It is capable of outputting true zero values (representational sparsity)

   Disadvantages:
   - Relu: Activation may blow up (not bounded). Exploding gradient.
   - Dying Relu / Hard Tangent the derivate might stay at 0 if the activation gets too low.

   Prevent:
   - Using leaky ReLU / Parametrical ReLU (there are more alternatives)
   - Use batch normalization and weight penalties (regularization)

Exercise 6. Transformers

1. What are the improvements of transformer-based language models (BERT, RoBERTa, ...) over ELMo?
2. Formally define the *multi-head self-attention* block and compare it to non-parametric self-attention. Assume scaled dot-product attention for both.
3. How does positional encoding work in BERT? Why do we need it? What are its advantages?

SOLUTION

1. - Uses the whole context at once (Instead of concatenating two unidirectional recurrent models), works better for distant relations
   - Higher level of abstraction (supports more layers, which theoretically can capture more abstract relations between words)
   - Multilingual
   - Uses word pieces instead of character embeddings
   - Training can be parallelized
2. $MultiHead(Q, K, V) = [head_1, \ldots, head_h]W_0$
   where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

   $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$

   Basic self-attention has no trainable parameters. In multi-head self-attention, a different set of weights is trained for the query, key and value transformation matrices for each head. To maintain the input length over multiple attention layers, multi-head self-attention must output smaller representations of the input and then concatenate them (For instance, for an input of 512 dimensions and 8 attention heads, each attention head output should have a length of 64.

3. A d-dimensional vector that encodes the wordpiece position in the sentence is added (not concatenated) to the d-dimensional vector representing the wordpiece. The positional vector uses sinusoidal functions with varying frequences so that when two positions are multiplied, the result is scales with the distance between such positions.

   We need positional encoding because the attention block is not recurrent and does not have any sense of position/order for each input.

   Advantages:

   - It is able to encode distances between inputs even for long sequences
   - Because it is added and not concatenated, less weights are required
   - They are precomputed and don't have to be trained

Annex

Absolute discount

$$P_{AD}(X = x) = \begin{cases} \frac{C(X=x)-\delta}{N} & \text{if } C(X = x) > 0 \\ \frac{(B-N_0)\delta/N_0}{N} & \text{otherwise} \end{cases}$$

Lidstone's Law

$$P_{LID}(X = x) = \frac{C(X = x) + \lambda}{N + B\lambda}$$

Jaccard similarity

$$sim_{jac}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{a}{a + b + c}$$

Cosine similarity

$$sim_{cos}(X, Y) = \frac{|X \cap Y|}{\sqrt{|X|} \cdot \sqrt{|Y|}} = \frac{a}{\sqrt{(a + b)}\sqrt{(a + c)}}$$

Parse Tree Probability

$$P(t) = \prod_{r \in t} q(r)$$