— Modeling and Transfer —

ÿ How to deal with the scarcity of labeled data? What is the role of language models?

1

Language modeling motivated by problem solving (recap)

q **How to model natural language?**

ÿ 2 main approaches

ÿ motivated by 2 types of problems with 2 types of resolutions

q **Grammar**

ÿ to answer the **problem of belonging:** ÿ is a given sequence a sentence of language L?

q **Language model**

ÿ to answer the **continuation problem:** ÿ what is the most likely continuation of a given sequence of the language L?

2

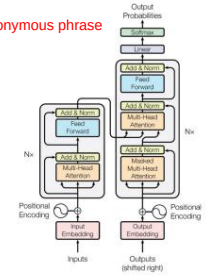
Þ Probabilistic neural modeling

q **Automatic translation**

ÿ Transformer originally designed for translation

automatic, from a phrase in a source language, obtain a synonymous phrase in a target language

ÿ eg “Cork is extracted from the cork oak.” -> “Cork is extracted from the cork oak.”



q **Output generation**

ÿ Made in time steps

ÿ “Cork is extracted from the cork oak.” -> “Cork”

ÿ “Cork is extracted from the cork oak tree. -> Cork” -> “is”

ÿ “Cork is extracted from the cork oak tree. -> Cork is” -> “extracted”

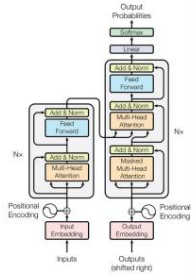
3

Continuation problem

q **Text generation**

ÿ Transformer can be reused to solve the continuation problem

ÿ From the same language, go from an input segment to the segment that can follow it



ÿ “Cork is extracted” -> "of"

ÿ “Cork is extracted -> from” -> “the“

ÿ “Cork is extracted -> from” -> “cork oak”

ÿ “Cork is extracted -> from the cork oak” -> “.”

4

Neural language model

q **Self-supervised learning**

ÿ to train a probabilistic neural language model of a given language, the training data are running ("raw") texts of that language

ÿ Simply (automatically) remove the segment that comes next and ask for your prediction from the previous segment

q **Colossal training datasets**

ÿ There is no need to annotate the training set! Just collect mass of texts from the web...

ÿ Much larger training datasets than (annotated) ones exist for any supervised task

Training (recap)

q **Forward passage**

ÿ **multi-class classification:** each item in the vocabulary is a class

ÿ network output layer/vector **y^** with vocabulary size

ÿ predicted item corresponds to the component y^[i] with the highest value

q **Loss**

ÿ Correct output: one-hot vector **y** = y[1] **…** y[m] with 1 component whose index is the index in the vocabulary of the next item, zero in the remaining

ÿ loss function, e.g. cross entropy **L(y^,y)** = – ÿi y[i] **log(y^[i])**

q **Backward passage**

ÿ gradient **L'(y^)** é L'(y^[i])=1 e L'(y^[j])=0 para qq j != i

Language model performance

q **Loss in individual prediction (recap)**

ÿ correct output: a one-hot vector **y** = y[1] **…** y[m] with 1 at index i of the next correct item (in the training data) and zero in the remaining indices

ÿ prediction: **y^** = y^[1] **…** y^[m] normalized by softmax: each component in [0,1] and add them all to 1

ÿ being cross entropy: **L(y^,y)** = – ÿi y[i] **log(y^[i])**

ÿ individual loss: - (0 x **log(y^[0])** + ...+ 1 x **log(y^[i])** + ÿ or - ··· + 0 x **log(y^[n])) log(y^[i])**

q **Model performance** (training completed)

ÿ for model tested on n+1 items, average loss:

ÿ – ÿj=0 log **(y^j) /** n

ÿ lower value is better performance: lower loss/error

Using language models

q **Text generation**

ÿ Answer questions

ÿ Fill out questionnaires

ÿ Create fake news

ÿ Dialogue

ÿ ...

q **Leverage supervised tasks**

ÿ with transfer learning

Þ Pre-train and tune

q **Pre-** train

ÿ **First phase of training:**

ÿ train the neural model as a language model ÿ **with colossal volumes of training data,** i.e. text

q **Fine** -tuning

ÿ **Second training phase:** ÿ

continue training the neural model to solve the supervised task of interest ÿ e.g. argument

extraction, emotion detection, inference,... ÿ with **annotated training data, i.e. of small volume**

Branco

9c.9

9

Linguistic regularities

q **Pre-training: the (general) language model** ÿ **First training phase:** ÿ learning "general" linguistic regularities

q **Fine-tune: for the relevant (specific) task** ÿ **Second training phase:** ÿ complement with learning of "specific" regularities of the task/application

ÿ called downstream tasks

Branco

9c.10

10

Tuning

q **Global**

ÿ all model weights are allowed to adjust during the period of tuning

q **Selective**

ÿ An extra layer is added to the model and only the weights of this extra layer are changed during tuning (in the encoder family – more on this later)

ÿ Or in some designated layers the respective weights are left to adjust

Branco

9c.11

11

Dealing with the scarcity of labeled data

q **Better performance**

ÿ **very advantageous compared to just training for the task of interest, i.e. with only (less) annotated training data**

ÿ **way to mitigate the scarcity of labeled data**

Branco

9c.12

12

Þ Modeling linguistic knowledge

q **Symbolic approach**

ÿ Linguistic regularities are (manually) specified in a grammar

ÿ Which solves the problem of belonging and provides representation of meaning based on inference

ÿ Which will serve as a base to be embedded and enable applications

q **Neural approach**

ÿ Linguistic regularities are (automatically) captured in a language model

ÿ Which solves the continuation problem and provides a representation of meaning based on vector space ÿ Which will serve as a basis for applications by complementary tuning of its weights

Brazzo

9c.13

13

Continuation tasks

q **Next item** (“next token”)

ÿ given a segment, predict which item should follow ÿ called causal language modeling

Brazzo

9c.14

14

Continuation tasks (with tokens)

q **Masked Language Modeling** (“MLM”) ÿ predict the words that have been masked or (randomly) replaced in a sentence ÿ e.g. for “I looked at my [MASK] and noticed that [MASK] was late.”, predict which words should replace each occurrence of [MASK]

ÿ eg for “I looked at my watch and realized she was delayed.”, predict which words should replace those that were exchanged

q **Token** deletion

ÿ predict the words that were deleted (randomly) in a sentence

Brazzo

9c.15

15

Continuation tasks (with sequences)

q **Prediction of the next sentence**

(“NSP – Next Sentence Prediction”) ÿ of two given sentences, indicate whether the second one can be the continuation of the first

ÿ eg After “Cork is extracted from the cork oak.” what follows? ÿ “Represents 10% of Portuguese exports.” ÿ eg After “The birds fly in the sky..” what follows? ÿ “Sailing is a water sport.”

q **Deform sentences**

(“sentence warping”) ÿ start with a random word from the sentence/document, put the remaining tokens that did not appear at the beginning of the sentence, at the end of the sentence

Brazzo

9c.16

16

Þ Transfer learning

q Tuning, after pre-training ÿ
particular case of transfer learning

q **Transfer learning**

("transfer learning") ÿ
knowledge learned in solving one problem is applied to solving another problem

17

Dealing with data scarcity

q **Lack of annotated data**

ÿ annotated data are scarce, difficult to obtain

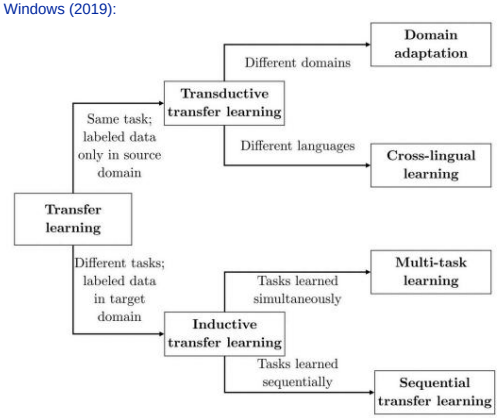
ÿ transfer learning is one of the techniques that aims to mitigate this difficulty

ÿ leveraging data from other tasks for the task of interest

18

Transfer taxonomy

Windows (2019):



19

Transductive learning by transfer (same task)

q **Domain adaptation** ÿ same task, eg

Named Entity Recognition ÿ
more annotated data in texts from one domain, eg news ÿ
but insufficient data for another domain, eg legal text ÿ apply the model
trained in the rich domain to the scarce domain

q **Aprendizagem multilingue** ("cross-lingual learning")

ÿ the same task, e.g. POS tagging ÿ more
annotated data in one language, e.g. English ÿ but insufficient
data for another language, e.g. Portuguese ÿ apply the model trained on
English texts to Portuguese

20

Inductive transfer learning (different tasks)

(tarefas diferentes)

q Multi **-task learning** ÿ same language

and domain, eg English and
blogging ÿ but different tasks, eg sentiment analysis
and trend detection
emotions

ÿ merge the two training data sets and do a single model training

q Sequential transfer **learning** ÿ same language and domain, e.g. English and

blogging ÿ but different tasks, e.g. language
modeling and NER ÿ training in two phases: first on one task; then
continue training, focusing on a second task

- Conclusão -

q Index

ÿ Neural language models

ÿ Pre-train and tune

ÿ Tuning

ÿ Continuation tasks

ÿ Transfer learning

How to adopt progress benchmarks? What variants of
Do Transformers exist? How to assemble the input and output data in
terms of different tasks?