NLP - state of the art

November 2021

Antoni Maciąg

Overview

- Important concepts, model classes and techniques used in SotA NLP.
- Key areas of NLP, as well as some smaller ones.
- Models that are SotA for the respective areas and their evaluation with appropriate metrics.
- Most important base models currently used.

Quick note on word representations (embeddings)

- We need vector representations of words (can't feed a string to a neural network).
- Simplest approach: one-hot vectors, obviously flawed but still used sometimes.
- Better idea: word embeddings. Works like a charm, preserves similarities and even complex relations. For details and learning algorithm see: word2vec word2vec-mikolov.pdf (udacity-data.com).

Another quick note on evaluation metrics

- Two important general notions in model evaluation: precision and recall.
- Intuitive explanation: Explaining precision and recall. The first days and weeks of getting… | by Andreas Klintberg | Medium.
- Related to binary classification tasks. Intuitively, recall tells us how many (proportionally) of the items that are positive have been classified as such. Precision tells us how many of the items judged to be positive are actually positive.
- High precision sounds good, but defined as it is, it doesn't guarantee protection from a large number of false negatives. However, high precision + high recall does.
- F1 = 2pr/(p+r) - very common evaluation metric.
- All of this can be generalized for non-binary tasks! We only need some similarity metric between expected and provided results.

Generally - important model classes in today's NLP

- Neural Networks maybe? :)
- Recurrent Neural Networks with improvements like LSTM
- Convolutional Neural Networks
- Seq2seq, Transformers

Recurrent Neural Networks

- In contrast to feed-forward NNs, connections between their neurons can form cycles and do not have to be organized into layers.
- Instead of taking one input and returning one output, they can take a sequence of inputs and return a sequence of outputs.
- While taking the next input, information about the previous inputs is ‘stored’ in the signals going around the network (hidden state).
- So the output depends on all the earlier inputs!
- But it does depend more heavily on the more recent ones, so it’s harder to learn long-range dependencies (small effect on current hidden state, so vanishing gradient problem).

LSTM

- Long-short term memory, introduced in the 90s to fix the vanishing gradients problem in RNNs.
- For a network of *n* neurons, the hidden state will be a vector of size *n*. Let's introduce another vector of size *n* for more memory and call it the *cell state*.
- Memory usage directed by gates: input, output, forget. Gates are *n*-vectors of values 0 through 1, dynamically computed for each timestep.
- SotA ca. 2014-2017 for tasks like handwriting/speech recognition/machine translation.

seq2seq

- Generic model: summarization, translation, dialogue.
- Two recurrent neural networks (or more complicated models): encoder and decoder.
- Encoder's hidden state after the final word (aka encoding) is decoder's first hidden state, the output is irrelevant. Hidden states for other words are also ignored in vanilla seq2seq.
- After encoding, we feed the decoder a START token and get a probability distribution for the first word of decoded sequence. We take argmax of that, feed it back, and repeat until an END token is suggested.

Attention

- Improving upon seq2seq.
- Instead of using the encoding, for each word of the output sequence that we are about to produce, we first calculate the probability distribution of which word in the input sequence we should focus on (‘attend to’) now. Then we take the weighted sum of all encoder’s hidden states according to that probability distribution and use it as the decoder’s hidden state. The rest remains more or less the same.
- Further improvement: self-attention.

Self-attention and multi-headed attention

- Self-attention - intuitively, for an input sequence, it tells us how much each word of that sequence ‘attends’ to each other word (probability distribution again).
- Multi-headed attention - in natural language, words can relate to each other in many ways: semantic, grammatical, syntactic …
- So let’s learn a separate attention function for each of these aspects!

Transformers

- Further improvement upon the encoder-decoder model.
- Introduced in the following, very well-known paper: [1706.03762] Attention Is All You Need (arxiv.org) in 2017. Since then (!) they have become SotA in a *lot* problems in NLP.
- Doesn't use RNNs; instead focuses *heavily* on attention.
- Encoder has multiple layers, each of which has a feed-forward NN and uses multi-headed self-attention, so that words can give each other contextual meaning.
- Decoder works similarly, but of course each word can only get attention from the previously translated words.
- Performance superior to RNNs + better parallelization, huge GPU gains. Size matters in NLP (performance often in proportion to log of number of features).

CNNs for NLP

- Standard convolution idea: take the vector representations of all the words in a sentence, apply a filter, and you get a feature. Take a few different filters, and you get a feature vector.
- Straightforward tool to account for word context.
- Information from the whole sentence is equally important (as opposed to RNNs, which pay more attention to the ending than to the beginning).
- Architecture sometimes resembles that for image processing.

Key areas to consider

- Language Modeling (not what you might think)
- Natural Language Generation
- Machine Translation
- Question Answering
- Sentiment Analysis
- Summarization

Some minor problems also worth looking at

- Coreference resolution
- Constituency Parsing
- Dependency Parsing
- Named Entity Recognition

Language modeling

- Predicting the next word (or character! for subword models) in a sequence, like predicting the next word of a sentence when typing on a mobile phone.
- Or more accurately: calculating the probability distribution over all known words, predicting how likely each one is to come next, given the sequence of all words that came before it.
- Standard evaluation metric: perplexity.
- Base for some more complex problems like Machine Translation or Natural Language Generation.

Perplexity

- Corpus - a large volume of text, like Wikipedia articles (each internally concise - not random unrelated sentences).
- Perplexity is defined for a model and a corpus. Bad news! But that's the norm.
- Informally: the closer the model's predictions are to what actually happens in the corpus, the better.
- So let's take the product of inverse probabilities of all consecutive words in the corpus as would be predicted by the model based on the previous words, and normalize. Why not.
- **Note: the lower the perplexity, the better!**

$$\text{Perplexity}(M) = M(s)^{-1/n}$$

$$= \sqrt[n]{\prod_{k=1}^{n} \frac{1}{M(w_k|w_0 w_1 \cdots w_{k-1})}}$$

SotA in Language Modeling

- This paper seems to take the biscuit: 1901.02860.pdf (arxiv.org) (Transformer-XL Large, Dai et al., 2018), but it is *still under review*
- It achieves perplexity of 21.8 measured on Google Billion Word Benchmark, while some others I found go as low as 15.8, but only on smaller corpora.
- Uses, among others, Transformers (duh) and self-attention.
- Recurrent Neural Networks also prominent in the field.

Natural Language Generation

- Follows easily from Language Modeling - we can repeatedly take the word that is suggested to be the most likely, like in Facebook threads for bored people, or do something more complicated (beam search).
- It is a subtask in MT, Summarization, Question Answering …
- These problems have their own metrics and SotA models, which we'll talk about in some more detail in a moment.
- Beam search: in short, we build a *k*-ary tree in which the children of each node are the *k* words suggested by the LM to be the most likely. Each path from root to leaf is a suggested sentence. We analyze somehow which sentence is the most probable and choose it.

NLG makes for nice clickbaits

METRO

NEWS... BUT NOT AS YOU KNOW IT

NEWS SPORT ENTERTAINMENT SOAPS MORE TRENDING

UK WORLD WEIRD TECH

Elon Musk's OpenAI builds artificial intelligence so powerful it must be kept locked up for the good of humanity

Jasper Hamill Friday 15 Feb 2019 10:06 am

272 SHARES

Elon Musk's scientists have announced the creation of a terrifying artificial intelligence that's so smart they refused to release it to the public.

Machine Translation

- As we know: taking a sequence of text in one natural language and producing a sequence in another language that is understood by humans to mean roughly the same.
- Standard evaluation metric: BLEU.

BLEU

- Without going into detail, it evaluates models based on how similar their translations are to that of humans (n-gram overlap).
- Usually we do not compare the whole translated text, but rather individual sentences, each against a set of translations proposed by human translators, and then average.
- The output is between 0 and 1, and the higher, the better (but values close to 1 are not even reached for human-made translations).
- Easy to implement, automated (once you have the translations).
- Has its limitations, for example it doesn't (explicitly) take coherence, grammatical correctness etc. into account.

SotA in Machine Translation

- DeepL Press Information | Setting Records! with BLEU of 45.9 (I mean 0.459, but for some reason everybody presents the BLEU score multiplied by 100).
- Algorithm not publicly known.
- What we do know: Convolutional Neural Networks, Attention again, and some clever division of the source sequence into fragments that can be translated individually.

Question Answering

- Given a passage of text and a question, the task of answering that question, providing all the necessary information and no unnecessary information.
- Also called simply ‘[Machine] Reading Comprehension’, and rightly so.
- Standard evaluation metric: SQuAD (2.0).

Squad 2.0

- Stanford Question Answering Dataset.
- A set of 10^5 questions related to Wikipedia articles. Importantly, half of these questions *cannot* be answered based on their respective article.
- The model is evaluated in a similar manner to BLEU: each (answerable) question has three human-generated, correct answers. The model's answers are rated by their similarity to those provided by humans. The higher the score, the better.

SotA in Question Answering

- Not the highest SQuAD score I was able to found (90.9), but very close (90.5), and the highest that came with a paper: [2001.09694v4] Retrospective Reader for Machine Reading Comprehension (arxiv.org).
- Note: models already outperform humans (~86 SQuAD), so let's get ready for an AI apocalypse :)
- This one uses a somewhat non-standard architecture: two Transformer-based encoder-decoder modules, one of which is responsible for something analogous to humans' skimming, and the other - for in-depth reading.
- Other models close to SotA also use the encoder-decoder scheme.

Summarization

- No, not of the presentation :)
- Given a passage of text, the task of creating a significantly shorter passage that retains most of the relevant information.

ROUGE

- Recall-oriented Understudy for Gisting Evaluation
- Similar to BLEU: overlap of *n*-grams between the summaries proposed by the model and humans, with analogous strengths and weaknesses.
- Comes in different flavors depending, among others, on *n*.

SotA in Summarization

- This paper: PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization (arxiv.org), ROUGE-2: 24.56
- Yet again: fancy architecture based on Transformers.

Sentiment Analysis

- Given a passage of text (e.g. restaurant review), determine whether it is positive or negative.
- Simple evaluation metric: accuracy as percentage of right guesses for a given test set.
- SotA - this model: 1906.08237.pdf (arxiv.org), accuracy up to 96%.
- Transformers with some tricks for pre-training.

Coreference Resolution

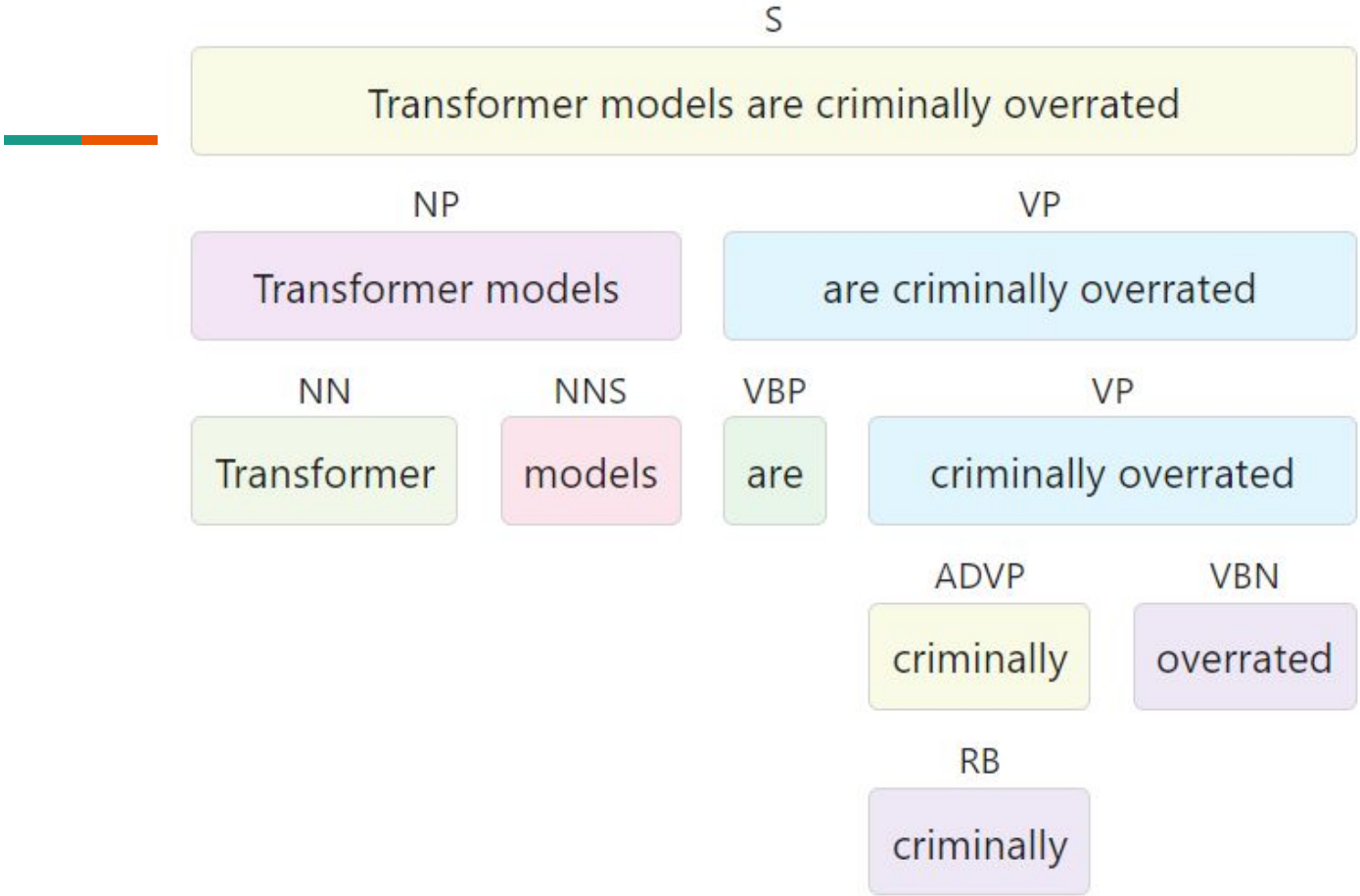
- Determining which words refer to the same entity. Simplest case: pronouns.
- ‘Thomas buys me flowers on our every anniversary’, said Theresa, bragging about her {her} boyfriend as she often does.
- Understanding that little pink ‘our’ proves to be a much harder task than you’d expect.
- Same goes for CR in general - it can’t be done well without a good understanding of how world works. A harder (!) alternative to Turing test was put forward based on this.
- Consider: ‘Pour beer from the bottle into the glass until it’s <*>’
- The reference behind ‘it’ depends on whether <*> means ‘empty’ or ‘full’.
- Standard evaluation metric based on precision and recall, for instance for this task: CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes - ACL Anthology

SotA in Coreference Resolution

- 2101.00434.pdf (arxiv.org)
- F1=80.3
- Guess what’s the main model class behind it :)

Constituency Parsing

- Dividing a sentence into parts that belong to a grammar category, like noun phrase, verb phrase, verb...


S
Transformer models are criminally overrated
NP
Transformer models
VP
are criminally overrated
NN
Transformer
NNS
models
VBP
are
VP
criminally overrated
ADVP
criminally
VBN
overrated
RB
criminally

Dependency Parsing

- Extracting relationships between the words in a sentence.
- For example: ‘I need a chocolate cookie.’
- ‘a’ is the article for ‘cookie’, ‘cookie’ is the object of ‘need’, ‘I’ is the subject of ‘need’, and so on.

SotA in Constituency and Dependency Parsing (!)

- This bad boy: Rethinking Self-Attention: Towards Interpretability in Neural Parsing (aclanthology.org).
- F1 score of 96.38 for CP.

Named Entity Recognition

- Recognizing the (pre-defined) types of entities mentioned in a passage of text.
- E.g. extracting person names, addresses, hours, organization names.
- SotA: LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention (aclanthology.org).
- F1 = 94.3
- Transformers with some improvements to self-attention.

Now for some important base models

- GPT-2
- GPT-3
- BERT

GPT-2

- Created by OpenAI, February 2019.
- Generative Pre-trained Transformer 2.
- Got very good at language modeling and became very versatile: translation, question answering, summarization…
- The text it generates can often be distinguished from that written by a human.

GPT-3

- June 2020.
- No substantial improvements in the model as compared to GPT-2, but a significantly increased number of parameters, resulting in much better performance.
- Often generates human-like text.
- Can also generate computer code from speech.

BERT

- Created by some guys at Google, 2018.
- Used in the Google search engine now.
- Transformers, architecture very similar to the originally proposed.
- Pretrained on Language Modeling and next sentence prediction.
- Question Answering, Sentiment Analysis…
- Widely cited and improved upon.
- Every paper I found (while doing research for this presentation) that outperformed BERT bragged about outperforming BERT.

Sources

- Stanford University: CS224N: NLP with Deep Learning course, winter 2019 (available on YouTube)
- Tracking Progress in Natural Language Processing | NLP-progress (nlpprogress.com)
- Various papers available at arXiv.org e-Print archive, mostly linked above
- Wikipedia :(