NMLM

NLP

Jeff Abrahamson

6 septembre 2020

Disclaimer

We have 13 minutes for 6 years.

This will simplify quite a lot. Maybe too much. The goal is to trace the evolution of techniques and to give hints about why.

I'm trying to avoid the non-essential stuff, like optimisation improvements that aren't core to why the state of the art advanced.

All the papers are on github.

Overview

What's changed:

- (Much) better performance
- Polysemy
- Pre-trained models

Overview

- $\leqslant 2014$
- 2014: word2vec and improvements
- 2018: ELMo
- 2019: BERT
- 2018–2020: GPT

Overview

- $\leqslant$ 2014
- 2014: word2vec and improvements
- 2018: ELMo
- 2019: BERT
- 2018–2020: GPT

Overview

Pre-trained models:

- Context-free (word2vec, FastText, GloVe)
- Contextual
  - Unidirectional (ELMo (I think))
  - Bidirectional (BERT, GPT-2,3 (I think))

The literature talks about uni- and bidirectional but isn't clear about which algorithms are which. This is my interpretation, and I might be wrong. I'm not an NLP specialist.

Before 2014

TF-IDF (classic example)

- 1-hot encoding to get high-dimensional vectors
- TF-IDF to scale (weight)

Goal: classify documents (or phrases).

Before 2014

TF-IDF (classic example)

- 1-hot encoding to get high-dimensional vectors
- TF-IDF to scale (weight)
- PCA or t-SNE etc. (if we want lower dimensionality)

Before 2014

TF-IDF (classic example)

- 1-hot encoding to get high-dimensional vectors
- TF-IDF to scale (weight)
- PCA or t-SNE etc. (if we want lower dimensionality)

What do we have?

- A space of likely significant words.
- Likely no good semantic knowledge.

Word2vec

- NN, but not deep: think auto-encoder or RBL
- Train on context: similar context $\Rightarrow$ similar embedding
- Want: distortion $\approx$ meaning

Word2vec

- NN, but not deep: think auto-encoder or RBL
- Train on context: similar context $\Rightarrow$ similar embedding
- Want: distortion $\approx$ meaning

Mikolov: trade complexity for efficiency $\Rightarrow$ learn bigger datasets.

Word2vec

- NN, but not deep: think auto-encoder or RBL
- Train on context: similar context $\Rightarrow$ similar embedding
- Want: distortion $\approx$ meaning

Mikolov: trade complexity for efficiency $\Rightarrow$ learn bigger datasets.

Uses: search, translation, ...

Tomas Mikolov et al., Efficient Estimation of Word Representations in Vector Space, ICLR Workshop, 2013.

Word2vec

Encode words at learning time.

$$(\mathbb{R}^N \to \mathbb{R}^n \qquad \text{for } N \approx 5 \times 10^5 \quad \text{and} \quad n \approx 300)$$

Problem: no polysemy

Word2vec

How does it work?

It's learning, so we're optimising something. What?

Word2vec

How does it work?

It's learning, so we're optimising something. What?

similarity of context

Word2vec

How does it work?

It’s learning, so we’re optimising something. What?

cosine distance for similar contexts

Word2vec

How does it work?

It's learning, so we're optimising something. What?

cosine distance for similar contexts

Look at words around the target word. Think of as an n-gram. We want words with similar context to project close together, words with different context (ideally) not to.

Word2vec

Interesting properties:

- Magnitude is related to importance
  - Too common: rarely learns large magnitude
  - Too rare: rarely grows large
  - So Goldilocks property (mid-range is just right)

Word2vec

Interesting properties:

- Magnitude is related to importance
  - Too common: rarely learns large magnitude
  - Too rare: rarely grows large
  - So Goldilocks property (mid-range is just right)

- Learns stop words. Because their contexts are uncorrelated, they optimise near zero.

Word2vec

How does it work?

It's not a single algorithm, but presented in multiple variants ("efficiency improvements"). Two notable elements (architectures) are

- CBOW (continuous bag-of-words) – context predicts word
- SG (skipgram) – word predicts context (SGNS = skipgram negative sampling)

Tomas Mikolov et al., Distributed Representations of Words and Phrases and their Compositionality, NIPS 2013.

Word2vec

How does it work?

- SGNS factorises a word-context PMI matrix (This is not the most important take-away, but to point out that the mathematical foundations are slowly being understood.)

PMI = pointwise mutual information

$$pmi(x; y) = \log\left(\frac{\mathbf{Pr}(x, y)}{\mathbf{Pr}(x)\,\mathbf{Pr}(y)}\right) = \log\left(\frac{\mathbf{Pr}(x \mid y)}{\mathbf{Pr}(x)}\right) = \log\left(\frac{\mathbf{Pr}(y \mid x)}{\mathbf{Pr}(y)}\right)$$

Sanjeev Arora et al., A latent variable model approach to word embeddings, Trans. Assoc. Comput. Linguistics 4: 385-399 (2016).

Word2vec

How does it work?

Output is a softmax: cost is proportional to number of classes (50K words).

Word2vec

How does it work?

Output is a softmax: cost is proportional to number of classes (50K words).

Approximate the softmax to reduce cost: hierarchical softmax: $O(N) \Rightarrow O(\log n)$.

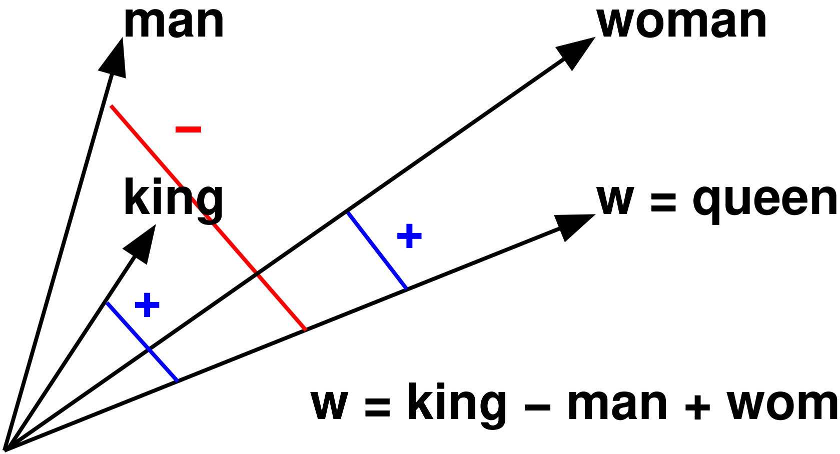
Yoav Goldber and Omber Levy, Word2vec Explained, arXiv 2014.

Word2vec

The famous “man : woman as king : queen” in maths: we’re maximising two similarities and minimising a dissimilarity.

But remember that we’re really working with cosine similarity.

Word2vec



w = king – man + woman

$$w\cdot\ \mathbf{king} - w\cdot\ \mathbf{man} + w\cdot\ \mathbf{woman}$$

Word2vec

More references (hardly exhaustive):

- *Yann LeCun, Bengio, Hinton, Deep Learning, Nature 521, 436–444(2015).*
- *Omer Levy and Yoav Goldberg, Neural Word Embedding as Implicit Matrix Factorization, 2014.*
- *Omer Levy and Yoav Goldberg, Linguistic Regularities in Sparse and Explicit Word Representations, 2014.*
- *Omer Levy et al., A Strong Baseline for Learning Cross-Lingual Word Embeddings from Sentence Alignments, 2017.*

Word2vec

Improvements:

1. FastText (Facebook)

   Piotr Bojanowski et al., Enriching Word Vectors with Subword Information, 2017.

   Armand Joulin, Bag of Tricks for Efficient Text Classification, 2016.

Word2vec

Improvements:

1. FastText (Facebook)
2. GloVe (“Global Vectors”)

Jeffrey Pennington et al., GloVe: Global Vectors for Word Representation, 2014.

`https://nlp.stanford.edu/projects/glove/`

Word2vec

Improvements:

1. FastText (Facebook)
2. GloVe (“Global Vectors”)
3. ULMFit (“Universal Language Model Fine Tuning”)

Jeremy Howard and Sebastian Ruder, Universal Language Model Fine-tuning for Text Classification, 2018.

Andrew Dai and Quoc Le, Semi-supervised Sequence Learning, 2015. *(← not ULMFiT, actually)*

Word2vec

Improvements:

1. FastText (Facebook)
2. GloVe ("Global Vectors")
3. ULMFit ("Universal Language Model Fine Tuning")

Better performance, similar properties.

What NLP was last quarter century, just a lot better.

Word2vec

Improvements:

1. FastText (Facebook)
2. GloVe (“Global Vectors”)
3. ULMFit (“Universal Language Model Fine Tuning”)

Better performance, similar properties.

What NLP was last quarter century, just a lot better.

Think twice before using any of these, they’re mostly obsolete.

ELMo

ELMo

- Encode words before and after (not just at learning time)
  - No: dictionary: map word to vector
  - **Yes:** on the fly, run context through deep network to produce new context
- Captures linguistic context: polysemy
- Models word use: captures both syntactic and semantic information
- Most tasks better than word2vec and relatives (question answering, named entity exceptions, sentiment analysis, . . .)

ELMo

- Bidirectional LSTM + residual connectors between 1st and second layer
- Character embedding rather than word: helps with out-of-vocabular words
- Convolutional filters instead of n-gram features

Up to here, similar to Jozfowicz

Rafal Jozfowicz, Exploring the Limits of Language Modeling, 2016.

ELMo

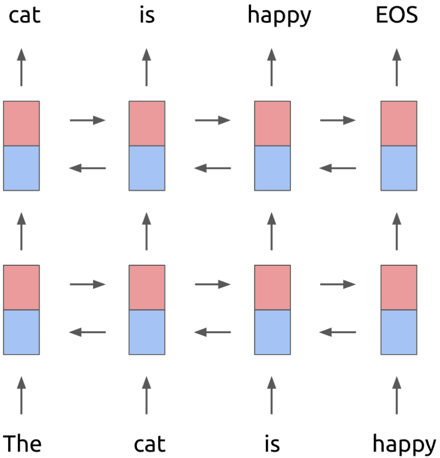
- Bidirectional LSTM + residual connectors between 1st and second layer
- Character embedding rather than word: helps with out-of-vocabular words
- Convolutional filters instead of n-gram features

- Learn different representations for different tasks
- Transfer learning

In CV, it's standard to learn ImageNet and transfer to problem at hand.

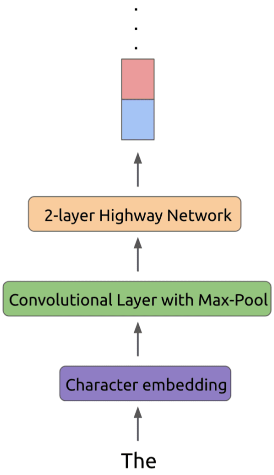
ELMo shows we can do this for NLP problems.

ELMo



https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/

ELMo



https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/

ELMo

See also

- *Matt Gardner et al., AllenNLP: A Deep Semantic Natural Language Processing Platform*
- *Yoon Kim, Character-Aware Neural Language Models, 2015.*
- *Matthew Peters et al., Deep contextualized word representations, 2018.*
- *Rupesh Kumar Srivastava et al., Highway Networks, 2015.*

BERT

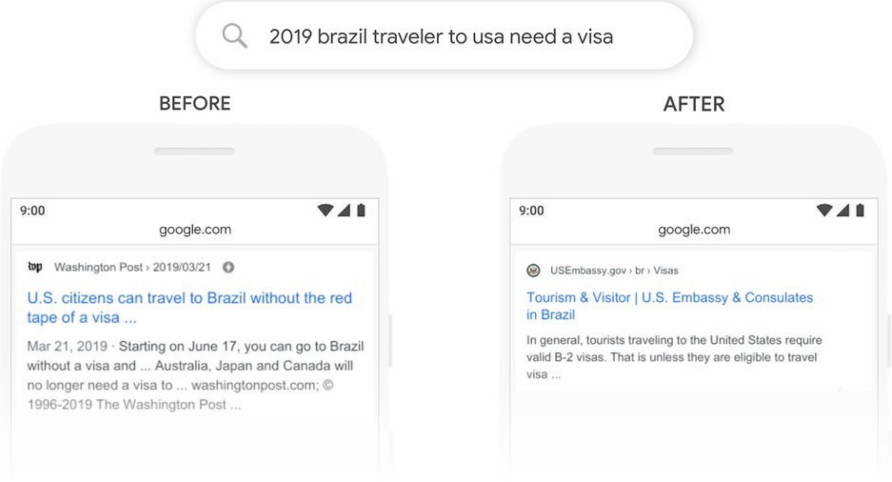
BERT

- Builds on ELMo and first GPT
- Encoder-decoder: use self-attention to encode and attention to decode

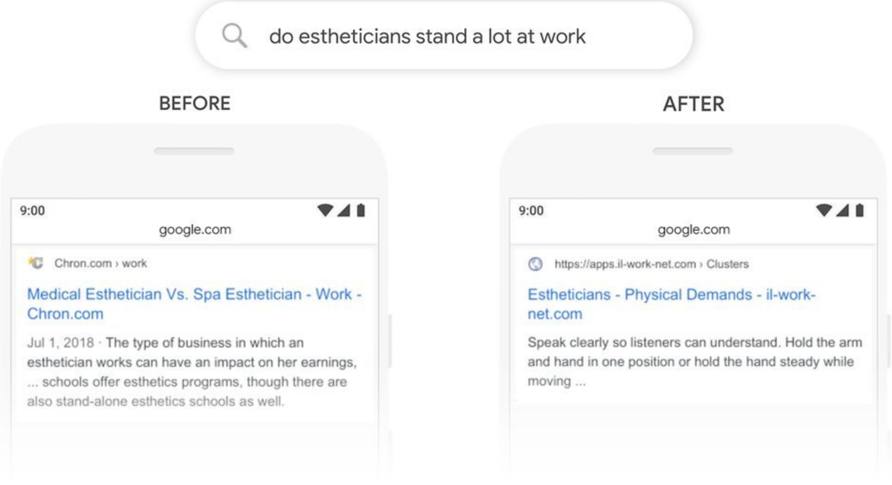
Jacob Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019.

Ashish Vaswani, Attention is All You Need, 2017.

BERT


2019 brazil traveler to usa need a visa
BEFORE
AFTER
9:00
google.com
Washington Post › 2019/03/21
U.S. citizens can travel to Brazil without the red tape of a visa ...
Mar 21, 2019 · Starting on June 17, you can go to Brazil without a visa and ... Australia, Japan and Canada will no longer need a visa to ... washingtonpost.com; © 1996-2019 The Washington Post ...
9:00
google.com
USEmbassy.gov › br › Visas
Tourism & Visitor | U.S. Embassy & Consulates in Brazil
In general, tourists traveling to the United States require valid B-2 visas. That is unless they are eligible to travel visa ...

BERT


do estheticians stand a lot at work
BEFORE
AFTER
9:00
google.com
Chron.com › work
Medical Esthetician Vs. Spa Esthetician - Work - Chron.com
Jul 1, 2018 · The type of business in which an esthetician works can have an impact on her earnings, ... schools offer esthetics programs, though there are also stand-alone esthetics schools as well.
9:00
google.com
https://apps.il-work-net.com › Clusters
Estheticians - Physical Demands - il-work-net.com
Speak clearly so listeners can understand. Hold the arm and hand in one position or hold the hand steady while moving ...

GPT

Generative Pre-trained Transformer

- The first GPT (6/2018) was in the ELMo/BERT world (*very* roughly). It demonstrated real-world knowledge acquisition and long-range dependencies.

GPT

Generative Pre-trained Transformer

- The first GPT (6/2018) was in the ELMo/BERT world (*very* roughly). It demonstrated real-world knowledge acquisition and long-range dependencies.
- GPT-2 and GPT-3 distinguished themselves by scaring people.

Questions?