Challenges in Transfer Learning in NLP

MADRID NLP MEETUP

29th May 2019

Lara Olmos Camarena

Ignacio Marrero Hervás

Contents

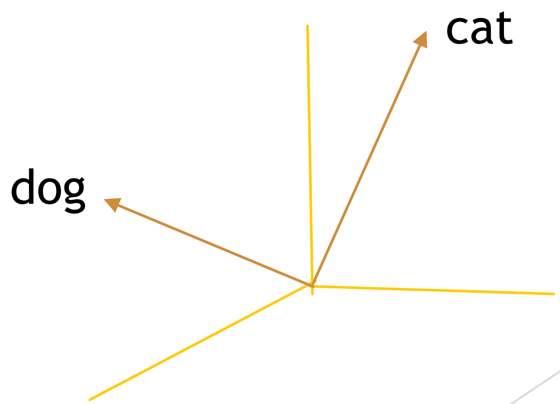
Introduction

Motivation

- *Natural language processing* systems are difficult to ***train***: ***word enrinchment*** and training meaningfull representation requires so much ***effort***.

- ***Machine learning*** *models* for *classification or clustering text* suffer with vectorial spaces (***VSM***) built with tokens (***bag of words***) because of it is ***high dimensional*** and ***sparse*** (*one-hot encoding*).
  - Loose of ***sequential information***.
  - Loose of ***relations between words***.

Solved with word embedding !



First definitions, challenges, frontiers

- ***Motivation: neural network models in NLP, frontiers and catastrophic forgetting***
  - A review of the recent history of natural language processing

    Neural language models (2001), multi-task learning (2008), Word embeddings (2013), seq2seq (2014),

    2018-2019: pretrained language models
  - ***Fighting with catastrophic forgetting in NLP***
  - *Frontiers of natural language processing*

- ***Transfer learning: formalization***
  - NLP's ImageNet moment
  - https://indico.io/odsc-2018-effective-transfer-learning-for-nlp/
  - Transfer learning with language models
  - Neural Transfer Learning for Natural Language Processing

Brief review of the state of art

- ***Language modelling,*** sometimes with ***neural network methods,*** to solve this problem: ***low dimensional vectors*** (***embeddings***) for textual units selected: *characters, words, phrases or documents.*

- ***Words embeddings*** obtained as output of different models and algorithms:
  - *Random: uniform, Xavier*
  - *Predictive models: word2vec (skip-gram, CBOW)*
  - *Count-based or coocurrences: GloVe*
  - *Deep neural network models* with different blocks trained: *CNN, RNN, LSTM, biLSTM*
  - *Recent and more complex ones: subword information FastText, biLSTM based ELMo*

- New perspectives in transfer learning in NLP:
  - Use ***pre-trained word embedding*** to initialize word vectors in embedding layers or other tasks
  - NEW! Use ***pre-trained language models*** to directly represent text. Prodigy (spaCy), Zero-shot learning for classifiers (Parallel Dots), Universal Sentence Encoder (Google), ULMFit, **BERT** and OpenAI Transformer

  to use later in more complex or ***task specific*** models/systems.

Word embeddings

Word embedding: typical definitions

Word embedding *is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much* ***lower dimension****.*

Methods to generate this mapping include neural networks, dimensionality reduction on the word co-occurrence matrix, probabilistic models, explainable knowledge base method, and explicit representation in terms of the context in which words appear.

Word and phrase embeddings, when used as the underlying input representation, have been shown to boost the performance in NLP tasks such as syntactic parsing and sentiment analysis.

(Ref.: Wikipedia)

A word embedding is a learned representation from text where words that have the ***same meaning*** *have a similar representation.*

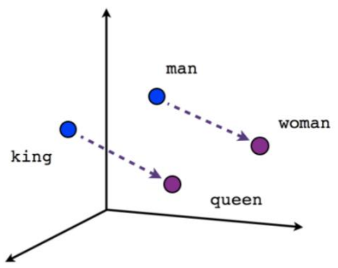
(Ref.: Machine Learning Mastering)

The ***Distributional Hypothesis*** *is that words that occur in the same contexts tend to have similar meanings (Harris, 1954). The underlying idea that "a word is characterized by the company it keeps" was popularized by Firth (1957), and it is implicit in Weaver's (1955) discussion of word sense disambiguation (originally written as a memorandum, in 1949). The Distributional Hypothesis is the basis for Statistical Semantics. Although the Distributional Hypothesis originated in Linguistics, it is now receiving attention in Cognitive Science (McDonald and Ramscar, 2001). The origin and theoretical basis of the Distributional Hypothesis is discussed by Sahlgren (2008).*

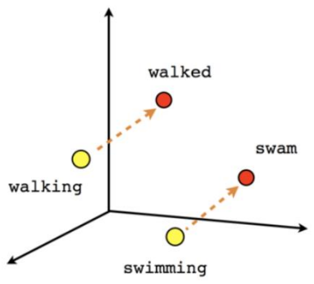
(Ref: ACL Wiki)

Pre-trained word embedding

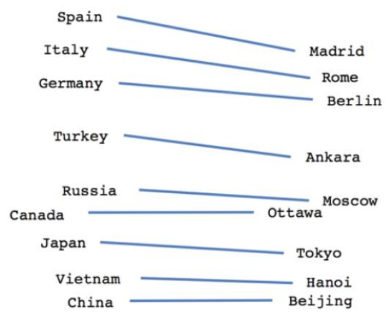
- **Word embedding** helps to **capture vocabulary** from a *corpus* (f.e.: *public* or *web corpora*) not seen in the task specific language or limited time training and used in general common language.
- The **representation** allows to establish *lineal relations* and capture *similarities* between words from *sintax and contexts* seen in the **language** and **domain** where the word embedding was trained. Be careful with **bias**!
- If pre-trained with public and external corpus, some words will be not recognized (***unknown word problem***), when fighting with **jargon** and **argots**!



Male-Female



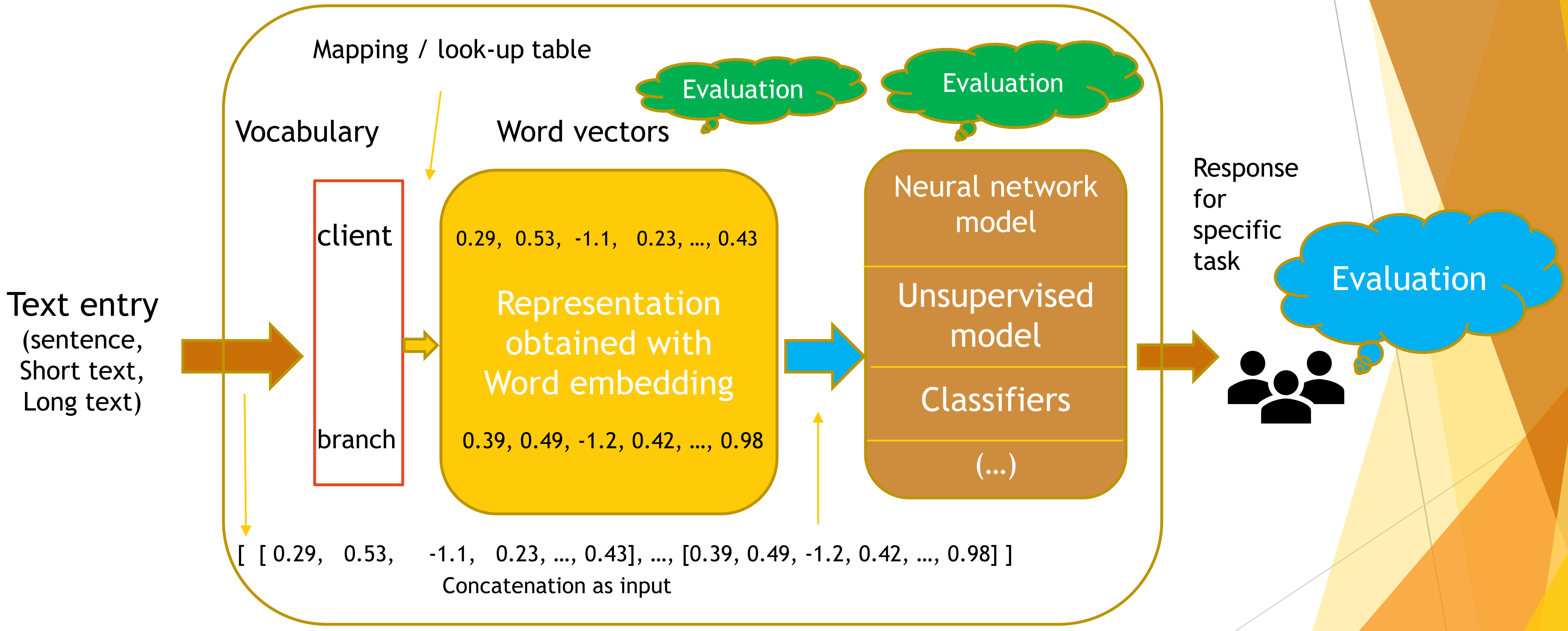
Verb tense



Country-Capital

Conceptual Architecture

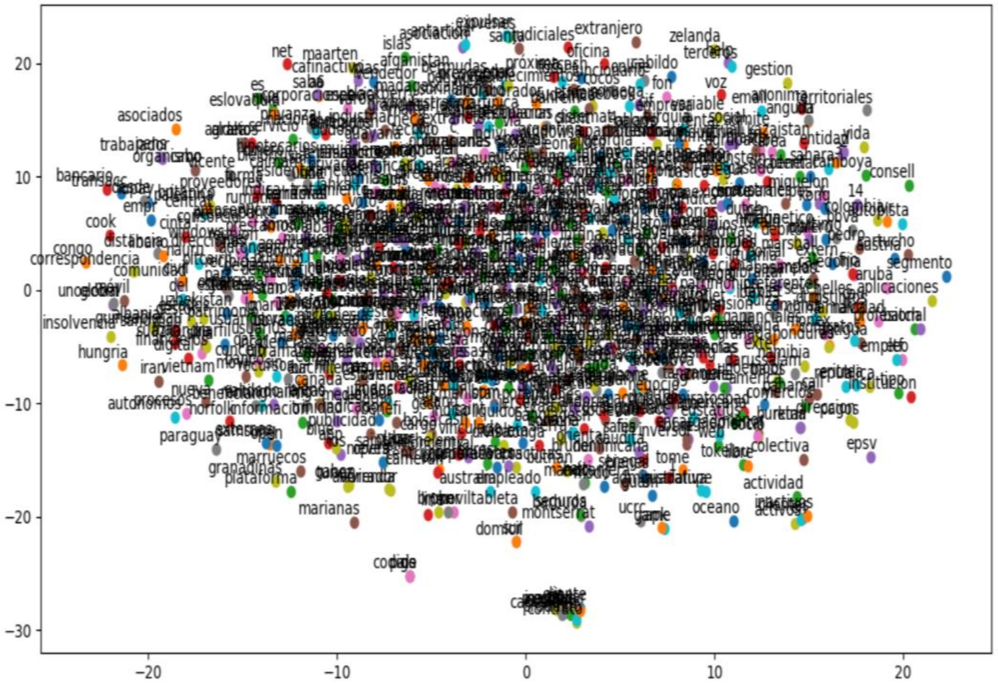
(common offline/training and online/predict/execution time)

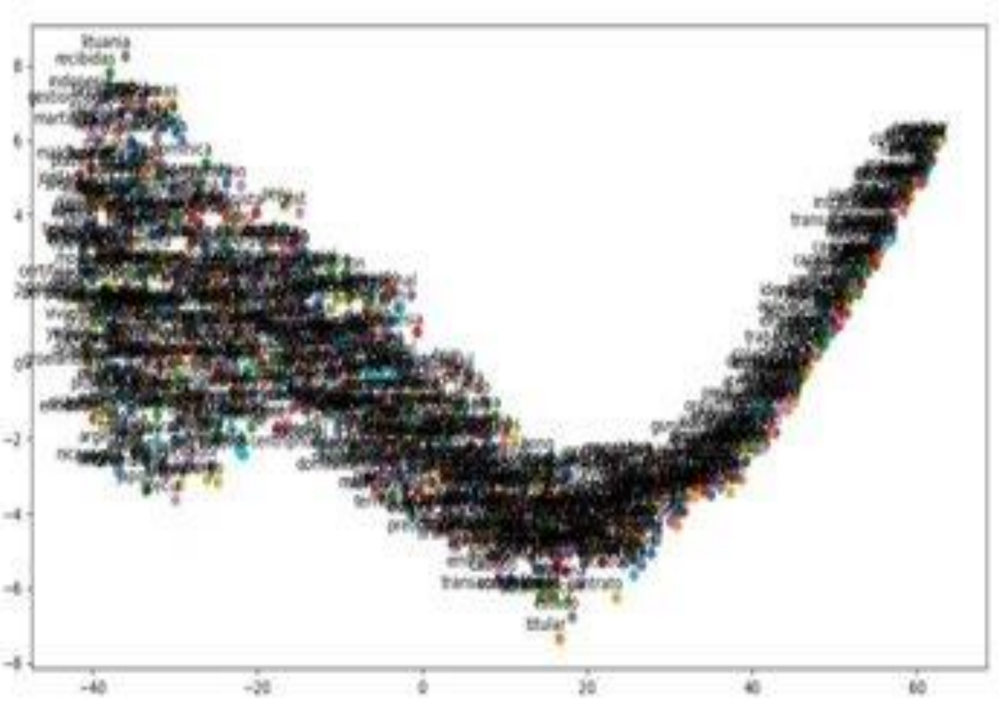


Training word embeddings

- Different Word embedding trained for *benchmarking* – perspective of pre-trained initialization of ***embedding layer*** for our neural network model
  - Initialization with **random distributions**: uniform, Xavier
  - Trained with **word2vec**
  - Trained with **GloVe**
  - Trained with **Tensorflow** with **NCE** estimation

- Using different ***corpora*** for training:
  - public corpora (example projects like Billion Words)
  - force-context phrases created for supervised synonyms
  - domain specific document corpus
  - real text users/industry dataset





GloVe example

Trained with force-contexts phrases built with vocabulary

Vocabulary size = 856, window size with = 3, vector size = 50

word2vec example

Corpus: force context phrases, Vocabulary size: 856 words
Algorithm: *word2vec*, **Word vector size**: 50

Evaluation of word embeddings

- ***Intrinsic proofs***: discovering internal word synonyms. *Synonyms trained, are still synonyms after embedding? Which embedding improves similarity for our task?*
  - ***Cualitative*** evaluation: word kneighbours clusters visualization (t-SNE and PCA), Brown clustering
  - ***Cuantitative*** evaluation: similarity test with cosine function and thresold using
- ***Extrinsic proofs***: testing Word embedding improvement when changing initialization for fix in specific task model.
  - Dataset, domain and task dependent
  - NLP processing pipeline must match the input sentence and word embedding construction (stopword removal, normalization, stemming...)

NEW! **Enso Benchmark** - https://github.com/IndicoDataSolutions/Enso

Conclusions

Thank you for your attention!

Any questions?

Appendix

Frameworks related

- *Gensim*: *https://radimrehurek.com/gensim/models/word2vec.html*
- *Glovepy*: https://pypi.org/project/glovepy/
- *SpaCy*: *https://spacy.io/*
- FastText – Facebook Research: https://github.com/facebookresearch/fastText
- Universal Sentence Encoder - Tensorflow:

  https://tfhub.dev/google/universal-sentence-encoder/2
- AllenNLP - ELMo - http://aclweb.org/anthology/N18-1202
  - https://allennlp.org/elmo
- Enso - https://github.com/IndicoDataSolutions/Enso
- BERT - https://github.com/google-research/bert

NLP&DL: Education

- **Stanford University** - Christopher Manning – Natural Language Processing with Deep Learning
  - https://www.youtube.com/watch?v=OQQ-W_63UgQ&list=PL3FW7Lu3i5Jsnh1rnUwq_TcylNr7EkRe6
  - http://cs224d.stanford.edu/
  - **http://web.stanford.edu/class/cs224n/**
- University of Oxford - Deep Natural Language Processing
  - **https://github.com/oxford-cs-deepnlp-2017/lectures**
- **Bar Ilan University's** - Yoav Goldberg - Senior Lecturer Computer Science Department NLP Lab - http://u.cs.biu.ac.il/~nlp/
  - *Book: Neural Network Methods for Natural Language Processing*
- Universitat Politècnica de Catalunya – Horacio Rodríguez - Embeddings working notes - http://www.cs.upc.edu/~horacio/docencia.html

References and papers

- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
- Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." *International conference on machine learning*. 2014.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning. "GloVe: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
- *Jurafsky, Daniel, & Martin, James H (2017). Speech and Language Processing. Draft 3º Edition. Chapter 15-16. Vector Semantics and Semantics with Dense Vectors*
- Goldberg, Yoav (2017). *Neural Network Methods for Natural Language Processing (Book)*
- Mnih, Andriy, and Koray Kavukcuoglu. "Learning word embeddings efficiently with noise-contrastive estimation." *Advances in neural information processing systems*. 2013.
- Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." *arXiv preprint arXiv:1801.06146* (2018).
- Schnabel, Tobias and Labutov, Igor (2015). *Evaluation methods for unsupervised word embeddings.*
- Wang, Yanshan and Lui, Sijia (2018). *A Comparison of Word Embeddings for the Biomedical Natural Language Processing*. arXiv:1802.00400v3
- *Ruder, Sebastian. On word embeddings.* http://ruder.io/word-embeddings-2017/, http://ruder.io/word-embeddings-1/
- *Rodríguez, Horacio. Embeddings, Deep Learning, Distributional Semantics en el Procesamiento de la lengua, ... ¿más que marketing?*
  - http://www.lsi.upc.edu/~ageno/anlp/embeddings.pdf, https://canal.uned.es/video/5a6fa2bcb1111f51708b4574?track_id=5a6fa2bdb1111f51708b4578