ITMO UNIVERSITY

NLP4IS

Unit: Word Embeddings – Research Topics (Part 2)

Oct 19th, 2018

Outline of this unit

- **Spacy**
- **Research projects**
- **Various aspects:** word similarity datasets, coref-resolution, relation extraction, etc.

Introduction to spaCy

- URL: https://spacy.io/
- Features (they claim):
  - Fast
  - Easy-to-use
  - Mature, gets things done
  - Also for industrial usage
  - Easy to integrate with deep learning
- Python-based

spaCy: First Steps

```
text = open('war_and_peace.txt').read()
doc = nlp(text)

# Find named entities, phrases and concepts
for entity in doc.ents:
    print(entity.text, entity.label_)


# Determine semantic similarities
doc1 = nlp(u'the fries were gross')
doc2 = nlp(u'worst fries ever')
doc1.similarity(doc2)


# Hook in your own deep learning models - or Coref
nlp.add_pipe(load_my_model(), before='parser')
```

spaCy: visual display

```
from spacy import displacy

doc_ent = nlp(u"'When Sebastian Thurn started working on self-driving
        cars at Google in 2007, few people outside of the company
        took him seriously.'")

displacy.serve(doc_ent, style='ent')
```

Examples at: **https://explosion.ai/demos/**

Show!

SpaCy: more functions

```
doc = nlp(u"Apple and banana are similar. Pasta and hippo aren't.")

apple = doc[0]
banana = doc[2]
pasta = doc[6]
hippo = doc[8]

assert apple.similarity(banana) > pasta.similarity(hippo)
assert apple.has_vector, banana.has_vector, pasta.has_vector, hippo.has_vector
```

NEW from here: Coreference resolution with spaCy

- Let’s look at some example code:

  https://github.com/gwohlgen/misc/blob/master/spacy_coref.ipynb

Exercise: spaCy test

- Take any input string
  - Sentence split with NLTK
- Make a spaCy document for every sentence
- Show the Named Entities in the doc
- Visualize the named entities of a sentence

Student Project: Coref resolution

- Based on the repository: https://github.com/gwohlgen/digitalhumanities_dataset_and_eval
- Basic goal:
  - Take the ASOIF and HP books
  - Split into short paragraphs
  - **Replace corefs with main mention**
  - Recompute word embedding models
  - **See how it affects results**

Student Project: Coref resolution

Details:

- Evaluate on different levels. Sentence, short paragraph, paragraph.
- Both book series
- Look at results, **manually evaluate correctness** (with B3 metric?!) of 100 sentences.
- **Provide stats:** how many corefs replaced, which changes in frequency …

Homework (unit 5)

- Do a very stupid co-ref system
  - Apply NER and POS tagging to the document
  - Any PER pronoun .. attach to the previous NER
  - Maybe think of options to make a bit better
- Evaluate with a couple of example sentences, eg from a book.
- Which **evaluation measure to use**? Come up with ideas and implement those measures
- **Present results**
- Run also **spacy / coref** on the text.
- Compare evaluation results of your system and spacy.

Presentation of Homeworks

- On 26th October, next unit
- **Homeworks:**
  1) Liubov (1st unit)
  2) Practice 2_IR/Homework unit 2 – IR
  3) Homework Unit 2 -- IR - Part 2 ???

     Not necessary – Bonus points if done.
  4) Practice 3/Homework (for unit 3 on 13th Oct): Word embeddings
  5) Practice 5/Unit 5 Homework -- Coref resolution

Project ASIOF/HP – Russian (2 students):

- **Russian language evaluation of basic ASOIF and HP datasets**
  - Find the books in Russian language
  - Preprocessing (together with Gerhard)
  - Model training (with various settings and methods (word2vec, Fasttext)
  - Translation of datasets! (careful)
  - Evaluation of datasets
  - Error analysis

Project “Sizes” (Overview):

- Project: Evaluate the accuracy of WE models trained on different text sizes
  - Inspired by: **Sahlgren and Lenci (2016)**: https://www.aclweb.org/anthology/D16-1099
  - Text Dataset: **Wikipedia**
  - Sizes: eg. 1M, 5M, 10M, 50M, 100M, 500M
  - **Evaluation Datasets:** Word similarity (MEN, WordSim-353, SimLex-999), Analogy datasets (Mikolov, BATS)
  - Here we specifically look at the **impact of term frequencies** (using frequency bins like in Sahlgren and Lenci) on task accuracy

The Effects of Data Size and Frequency Range on Distributional Semantic Models

Magnus Sahlgren
Gavagai and SICS

Alessandro Lenci
University of Pisa

pi.it

Abstract

This paper investigates the effects of data size and frequency range on distributional semantic models. We compare the performance of a number of representative models for several test settings over data of varying sizes, and over test items of various frequency. Our results show that neural network-based models underperform when the data is small, and that the most reliable model over data of varying sizes and frequency ranges is the inverted factorized model.

Sahlgren and Lenci, EMNLP, 2016 (1)

- Basic research question:
  - Effect of **data size** and **term frequency** in semantic models. Which **model to choose** if small data, or if low frequencies?
- Which datasets for evaluation?
  - Similarity and Vocabulary datasets:
    SimLex-999, MEN, Stanford Rare Words (RW)
    TOEFL synonyms and ESL synonyms
- Model types compared:
  - Word embeddings (word2vec SGNS and CBOW)
  - Matrix models (cooc, PPMI)
  - Factorized Matrix methods (SVD,..)

Sahlgren and Lenci, EMNLP, 2016 (2)

- What experimental setup? Which dataset sizes, which evaluation datasets?
  - ukWaC corpus 1.6 billion words after tokenization and lemmatization
  - Creation of subcorpora:
    First 1M, 10M, 100M, 1B words
  - Training settings: word window = 2 (very small)

Sahlgren and Lenci, EMNLP, 2016 (3)

- **Eval setup (continued): Frequency bins:**

  High-freq: 17K+,
  Med: 730-16K,
  Low: >729.

- **3 datasets** with term pairs, plus one MIXED (if terms in diff. categories).

  High: 1,387 , Mid: 656, Low: 350.
  Mixed: 3,458

Big Exercise:

- Look at **WordSim-353** dataset
  - http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/
  - Take combined.csv
- Compare the scores in WordSim-353 and the word2vec
  - Use **any kind of evaluation metric you wish** to compare against the gold standard
  - (One stupid idea: group into *n* **bins**, and measure set overlap)
  - If spaCy models not installed, use word2vec
  - https://github.com/gwohlgen/misc/blob/master/text8_30K.vec.zip

Sahlgren and Lenci, EMNLP, 2016 (4)

- Main results:
  - Neural models bad for very small data set, catching up at around 10/100M tokens. But CBOW good for low freq (strange).
  - For low frequencies: CBOW (27), PPMI (25.5), SG (19) -- surprising! For high-freq: SGNS best, surprising.

Project “Sizes” Description

- Sahlgren and Lenci evaluate on an 1B word corpus only
- We want to evaluate on smaller **corpora**.
    - N=1M, 5M, 10M, 50M, 100M, 1B (number of words)
    - We can use the same corpus, UkWaC.
    - Just take the first N words (like Sahlgren and Lenci)
- WE settings. We can make it the same (window = 2), but also want to try window = 5
- Sahlgren and Lenci: only similarity. Maybe we add analogy (to discuss)

Project "sizes" Description (cont'd)

- Input:
  - Plain-text corpus (provided by Gerhard)
  - Similarity datasets like WordSim253, SimLex999, MEN
  - Analogy datasets like (Mikolov/Google, BATS) ?
  - https://aclweb.org/aclwiki/Analogy_(State_of_the_art
)
  - use this? https://github.com/kudkudak/word-embeddings-benchmarks

Project “sizes” Description (cont’d - 2)

Steps:

(Remark: Highly recommend to use Python/Gensim for implementation -- it's proven as easy-to-use)

simi_voc = Get terms from Sim datasets (create a set / list of terms, to discuss)

Iterate over corpus sizes (5M, 10M, .. see above):

- Create corpus by picking sentences from the whole corpus to get to the required corpus size
- Compute word frequency bins for given corpus (size): LOW, MED, HIGH (Gerhard will explain)

Project “sizes” Description (cont’d - 2)

Steps (2):

(inside the same loop)

- Iterate over word embedding methods (word2vec-SGNS, FastText) and hyperparameter settings (to discuss):

  Generate embedding model

  Evaluate embedding model

  Store results.

Finally: analyse results results

Datasets: Wordsim, SimLex

- **WordSim-353:**
  - http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/
  - Quite old, from 2002 or so
  - Relatedness
- **SimLex-999:**
  - Harder than Wordsim-353, cause it describes similarity, not relatedness
  - But also includes a relatedness score
  - 666 Noun-Noun pairs, 222 Verb-Verb pairs and 111 Adjective-Adjective pairs.

Datasets: MEN

MEN

- https://staff.fnwi.uva.nl/e.bruni/MEN
- Newer, 2012
- two sets of English word pairs (one for training and one for testing) together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk via the CrowdFlower
- Relatedness and similarity not distinguished
- 3,000 word pairs, randomly selected from words that occur at least 700 times in the freely available ukWaC and Wackypedia corpora combined
- With and without POS-tags

Evaluation tool

- **“Word Embeddings Benchmarks”**
  - https://github.com/kudkudak/word-embeddings-benchmarks
- Looks like it’s very easy to evaluate models with that tool
- **Claims**:
  - 18 popular datasets
  - 11 word embeddings (word2vec, HPCA, morphoRNNLM, GloVe, LexVec, ConceptNet, HDC/PDC and others)
  - methods to solve analogy, similarity and categorization tasks
  - scikit-learn API and conventions

Project “sizes RU” Description – Evaluation of Russian language models

- Similiar to project 1, but for Russian language
- Russian language similarity datasets:

  https://github.com/nmrksic/LEAR/blob/master/evaluation/ws-353/wordsim353-russian.txt
  http://www.leviants.com/ira.leviant/MultilingualVSMdata.html
  http://rusvectores.org/static/testsets/ru_simlex965_tagged.tsv

  → WordSim353 and SimLex999 exist also for Russian

  → there is a corrected version: SimLex965
- Embeddings: Baselines pre-trained FastText, and RusVectores vectors
- Train corpora on different sizes of Russian Wikipedia or RNC, have different frequency bins
- https://github.com/rspeer/wiki2text
- https://github.com/attardi/wikiextractor
- https://dumps.wikimedia.org/ruwiki/latest/

Project “Sizes - RU” Description – Evaluation of Russian language models (cont’d)

- Train corpora on different sizes of Russian Wikipedia or RNC, have different frequency bins

  https://github.com/rspeer/wiki2text

  https://github.com/attardi/wikiextractor

  https://dumps.wikimedia.org/ruwiki/latest/

Project “OpenNRE” Relation Extraction - Topics

- Starting from a dataset created via DBpedia, apply a deep learning relation extraction toolkit
- Use the OpenNRE toolkit (or any other)
- Potential Topic 1:
  - Mostly about tuning, error analysis, etc with an existing dataset
- Potential Topic 2: create your own little dataset from DBpedia
  - Dataset creation
  - Evaluation with OpenNRE
- Or your own ideas – this whole block is a bit more experimental, and for adventurous students :)

Relation Extraction

- Open vs Closed.
- extraction of relation triples, in the form of

  (subject, predicate,object).
- Often relations between entities.
- Look at some text examples.

Relation Extraction – in our project

- Distant supervision in ML – what is it?
- Input data source: DBpedia
- What is DBpedia?

DBpedia

- DBpedia: large-scale extraction information/knowledge from Wikipedia infoboxes (at least in its early versions)
- Semantic Web format (RDF) – what is it?
- http://wiki.dbpedia.org/about
- http://dbpedia.org/page/Saint_Petersburg
- Problem: often messy

DBpedia (2)

- For text / NLP part we look at DBpedia abstracts, via the `dbo:abstract` property.
- DBpedia Textext Challenge
  - https://wiki.dbpedia.org/textext
  - In short: It is about the extraction of facts, eg. relations, from the Wikipedia text → with the long-term goal to increase the data available in DBpedia

Techniques for Relation Extraction

- Pattern-based
- ML-based
  - Classical: feature extraction, then apply ML algorithms like Naive Bayes, SVM, …
  - End-to-end with Deep Learning: skip feature extraction, and provide the sentence and the extraction relation pair as input.

OpenNRE

- https://github.com/thunlp/OpenNRE
- open-source framework for neural relation extraction
- TensorFlow-based
- Encoders: CNN, PCNN, RNN, BiRNN
- Various selectors and classifiers
- **Features:**
  - JSON data support.
  - Multi GPU training.
  - Validating while training.

OpenNRE / 2

- JSON Data format

```
[
    {
        'sentence': 'Bill Gates is the founder of Microsoft .',
        'head': {'word': 'Bill Gates', 'id': 'm.03_3d', ...(other information)},
        'tail': {'word': 'Microsoft', 'id': 'm.07dfk', ...(other information)},
        'relation': 'founder'
    },
    ...
]
```

- We provide all files already in that formats

Project Topics - Overview

1) **Russian language** ASOIF and HP experiments (2 students)
2) How do **various corpus sizes** of English text, and following different **frequencies of terms** affect WE accuracy (with given word similarity and analogy datasets?! Inspired by [Sahlgren and Lenci 2016] (2 students)
3) **Apply Co-ref Resolution** to ASOIF and HP books, measure the impact on results (2 students)
4) Relation Extraction with OpenNRE (or some other toolkit) with DBpedia data (distantly supervised)
5) Similar to project 2, but for **Russian language**.

Selection of Projects

- Also on 26th October
- **Goal:**
  - **Mini research project, clearly defined task**
  - If results are very nice, write a small arxiv.org paper, maybe even try to publish at some conference (writing done by Gerhard mostly)
  - **Goal for students:**

    Solid implementation

    Solid evaluation of results

ITMO UNIVERSITY

Thank you!

Questions?

Nov 2017