Современные дистрибутивно-семантические модели и их применение в лингвистических исследованиях
День 2

Андрей Кутузов
Университет Осло
Группа лингвистических технологий

Школа компьютерной лингвистики
Тюмень, февраль 2018

Содержание

Representing documents

- Distributional approaches allow to extract semantics from unlabeled data at word level.
- But we also need to represent variable-length documents!
  - for classification,
  - for clustering,
  - for information retrieval (including web search).

Representing documents

- Can we detect semantically similar texts in the same way as we detect similar words?
- Yes we can!
- Nothing prevents us from representing sentences, paragraphs or whole documents (further we use the term 'document' for all these things) as dense vectors.
- After the documents are represented as vectors, classification, clustering or other data processing tasks become trivial.

Can we do without semantics?

Bag-of-words with TF-IDF

A very strong baseline approach for document representation, hard to beat by modern methods:

1. Extract vocabulary V of all words (terms) in the training collection consisting of N documents;
2. For each term, calculate its document frequency: in how many documents it occurs (df);
3. Represent each document as a sparse vector of frequencies for all terms from V contained in it (tf);
4. For each value, calculate the weighted frequency wf using term frequency / inverted document frequency (TF-IDF):
    - $\mathrm{wf} = (1 + \log_{10}\mathrm{tf}) \times \log_{10}(\frac{\mathrm{N}}{\mathrm{df}})$
5. Use these weighted document vectors in your downstream tasks.

Representing documents

Bag-of-words problems

Unfortunately, simple bag-of-word does not take into account semantic relationships between linguistic entities.
No way to detect semantic similarity between documents which do not share words:

- California saw mass protests after the elections.
- Many Americans were anxious about the elected president.

It means we need more sophisticated semantically-aware distributed methods, like neural embeddings.

Содержание

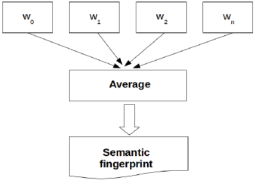
Distributed models: composing from word vectors

- Document meaning is composed of individual word meanings.
- Need to combine continuous word vectors into continuous document vectors.
- It is called a composition function.

Semantic fingerprints

- One of the simplest composition functions: an average vector $\vec{S}$ over vectors of all words $w_0 \ldots n$ in the document.
- We don't care about syntax and word order.
- If we already have a good word embedding model, this bottom-up approach is strikingly efficient and usually beats bag-of-words.
- Let's call it a 'semantic fingerprint' of the document.
- It is very important to remove stop words beforehand!

Distributed models: composing from word vectors



$$\vec{S} = \frac{1}{n} \times \sum_{i=0}^{n} \vec{w_n} \tag{1}$$

- You even don't have to average. Summing vectors is enough: cosine is about angles, not magnitudes.
- However, averaging makes difference in case of other distance metrics (Euclidean distance, etc).
- Also helps to keep things tidy and normalized (representations do not depend on document length).

Distributed models: composing from word vectors

Advantages of semantic fingerprints

- Semantic fingerprints work fast and reuse already trained models.
- Generalized document representations do not depend on particular words.
- They take advantage of 'semantic features' learned during the model training.
- Topically connected words collectively increase or decrease expression of the corresponding semantic components.
- Thus, topical words automatically become more important than noise words.

See more in [Kutuzov et al., 2016].

Distributed models: composing from word vectors

But...

However, for some problems such compositional approaches are not enough and we need to generate real document embeddings.

But how?

Содержание

Distributed models: training document vectors

Paragraph Vector

- [Le and Mikolov, 2014] proposed Paragraph Vector;
- primarily designed for learning sentence vectors;
- the algorithm takes as an input sentences/documents tagged with (possibly unique) identifiers;
- learns distributed representations for the sentences, such that similar sentences have similar vectors;
- so each sentence is represented with an identifier and a vector, like a word;
- these vectors serve as sort of document memories or document topics.

Not much evaluated (however, see [Hill et al., 2016] and [Lau and Baldwin, 2016])

Distributed models: training document vectors

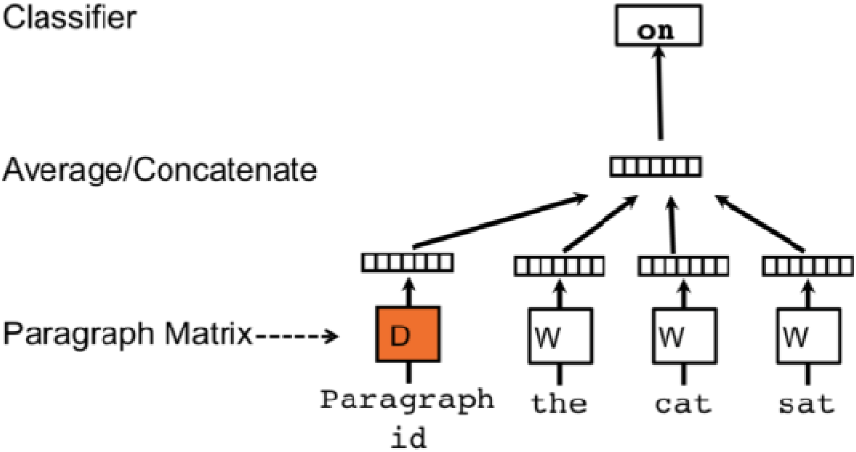
Paragraph Vector (aka doc2vec)

- implemented in Gensim under the name doc2vec;
- Distributed memory (DM) and Distributed Bag-of-words (DBOW) methods;
- PV-DM:
  - learn word embeddings in a usual way (shared by all documents);
  - randomly initialize document vectors;
  - use document vectors together with word vectors to predict the neighboring words within a pre-defined window;
  - minimize error;
  - the trained model can inference a vector for any new document (the model remains intact).
- PV-DBOW:
  - don't use sliding window at all;
  - just predict all words in the current document using its vector.

Contradicting reports on which method is better.

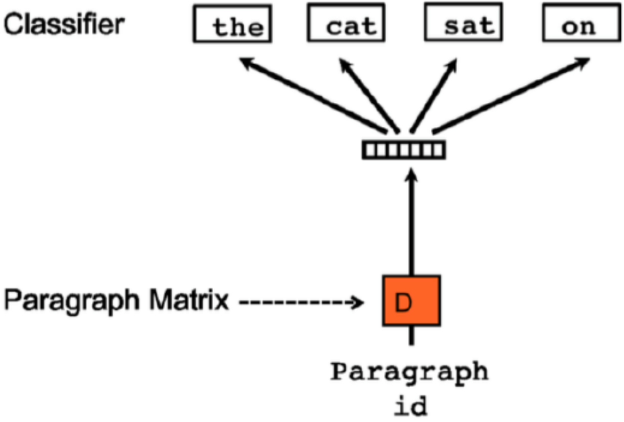
Distributed models: training document vectors

Paragraph Vector - Distributed memory (PV-DM)



Distributed models: training document vectors

Paragraph Vector - Distributed Bag-of-words (PV-DBOW)



Distributed models: training document vectors

Paragraph Vector (aka doc2vec)

- You train the model, then inference embeddings for the documents you are interested in.
- The resulting embeddings are shown to perform very good on sentiment analysis and other document classification tasks, as well as in IR tasks.
- Very memory-hungry: each sentence gets its own vector (many millions of sentences in the real-life corpora).
- It is possible to reduce the memory footprint by training a limited number of vectors: group sentences into classes.

Спасибо за внимание!
Вопросы?

Современные дистрибутивно-семантические модели и их применение в лингвистических исследованиях
День 2

http://rusvectores.org

Андрей Кутузов (andreku@ifi.uio.no)
Language Technology Group
University of Oslo

References I

Hill, F., Cho, K., and Korhonen, A. (2016).
Learning distributed representations of sentences from unlabelled data.
In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1367–1377. Association for Computational Linguistics.

Kutuzov, A., Kopotev, M., Sviridenko, T., and Ivanova, L. (2016).
Clustering comparable corpora of russian and ukrainian academic texts: Word embeddings and semantic fingerprints.
In Ninth Workshop on Building and Using Comparable Corpora, page 3.

References II

Lau, J. H. and Baldwin, T. (2016).
An empirical evaluation of doc2vec with practical insights into document embedding generation.
In Proceedings of the 1st Workshop on Representation Learning for NLP, pages 78–86. Association for Computational Linguistics.

Le, Q. and Mikolov, T. (2014).
Distributed representations of sentences and documents.
In International Conference on Machine Learning, pages 1188–1196.