INF5820
Distributional Semantics: Extracting Meaning from Data
Lecture 6
What's going on: recent advances and trends in the word embeddings world

Andrey Kutuzov
andreku@ifi.uio.no

30 November 2016

Contents

Hot topics in the distributional semantics world

Some aspects of meaning are problematic

- Detecting hyponyms, hypernyms and antonyms:
- they appear in similar contexts, but...
- cannot be replaced by each other:
- their paradigmatic relations are complex.
- Solutions:
  - integrating lexical contrast [Nguyen et al., 2016]
  - integrating syntactic paths [Shwartz et al., 2016]
  - etc.

Hot topics in the distributional semantics world

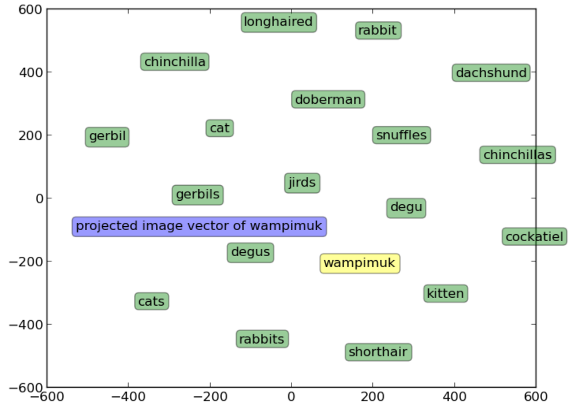
Some aspects of meaning are problematic

- Distributional models are not aware of implicit knowledge:
  - sky is blue
  - bananas are yellow
  - violins are brown.
- The answer is 'grounding':
- integrate language and vision.
- Aligning image embeddings with word embeddings.

Hot topics in the distributional semantics world

[Lazaridou et al., 2014]

Hot topics in the distributional semantics world



[Lazaridou et al., 2014]

Hot topics in the distributional semantics world

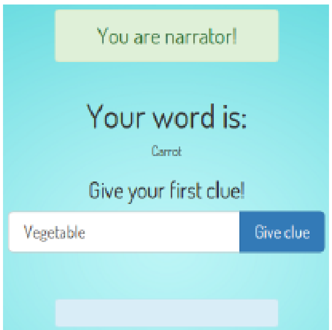
There is more than one language in the world

- Can we train bilingual or multilingual distributional models?
- We can!
- Lots of approaches emerged in the last 3 or 4 years.
- Thorough review of cross-lingual word embeddings in [Upadhyay et al., 2016]

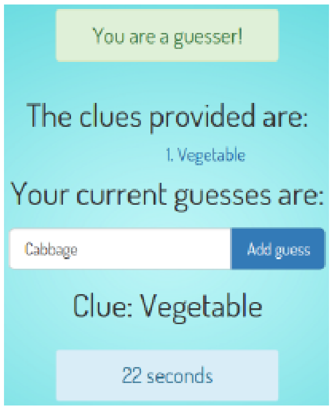
Hot topics in the distributional semantics world

How can we evaluate our models better?

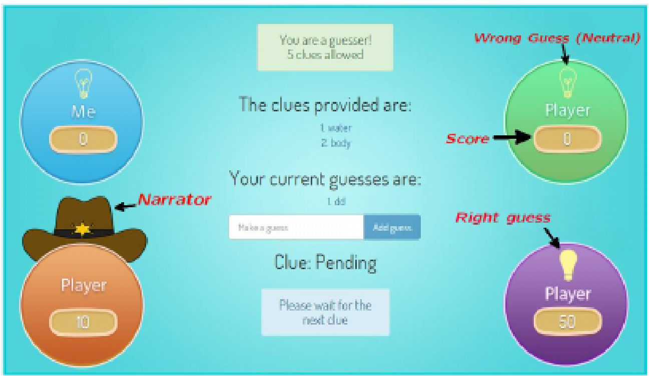
Generate new and more natural gold standard datasets!
Perhaps, using crowd-sourcing and gamification.



http://comp3096.herokuapp.com/
[Parasca et al., 2016]



http://comp3096.herokuapp.com/
[Parasca et al., 2016]



http://comp3096.herokuapp.com/
[Parasca et al., 2016]

Round	Narrator's clue	Guesser 1	Guesser 2
1a	fruit		
1b		orange	apple
2a	yellow		
2b		lemon	**banana**

Table 1: Successful game in 2 rounds for `banana`

Round	Narrator's clue	Guesser 1	Guesser 2
1a	rain		
1b		sun	jacket
2a	sunny		
2b		cloudy	windy
3a	noun		
3b		cloud	umbrella

Table 2: Unsuccessful try (3 rds., `weather`)

http://comp3096.herokuapp.com/
[Parasca et al., 2016]

Contents

Discussion of the obligatory assignment

- Good news: everyone has passed :-)
- What was interesting?
- Won't comment on purely pythonic issues, read the feedback.

Discussion of the obligatory assignment

- No need to include large data files in your submission
- Task 1: what is missing in *Semantic Vectors* web service?
- Some pointed they miss vector algebra (addition and subtraction)
- It's already there: see the *Calculator* tab (`http://ltr.uio.no/semvec/en/calculator`)

Discussion of the obligatory assignment

- Task 2 (evaluation)
- Very frequent issue:
- while calculating *SimLex999* correlation, you ignore (skip) out-of-vocabulary words
- Seems logical, but can be dangerous:
- imagine the model doesn't know 95% of the words from the dataset but is good in ranking the remaining 5%
- Can we say this model is perfect?
- Might be safer to produce similarity=0 for such word pairs (pretend the model thinks they are not related).

A good point: values of performance in *Google Analogy* test and in *SimLex999* test are not directly comparable ($64 > 34$ means nothing).

Discussion of the obligatory assignment

- Task 3 (document classification)
- Everyone used semantic fingerprints (as expected).
- *Gensim* model vector size can be retrieved with *model.vector_size*;
- Word vectors are *Numpy* arrays;
- Work with them using *Numpy* functions;
- Try not to mix with other data types.

Discussion of the obligatory assignment

- If you iteratively update your document vector (fingerprint):
- create it as a *Numpy* array from the very beginning:
- *numpy.zeros(model.vector_size)*
- then successively add word vectors to this array.

- Another way: first generate a zero matrix (*words number* X *vector size*);
- successively fill in the rows with word vectors;
- Then do *numpy.sum()* by axis 0 and *numpy.average()*;
- NB: do not try to expand the matrix (add new rows with new words)!
- Array expansion is comparatively slow in *Numpy*.

Discussion of the obligatory assignment

Interesting issue with initialization, leading to *Infs*

- You have a new document, you initialize the empty fingerprint variable with the vector of the first word:
- *fingerprint = model[first_word]*
- and continue updating it with the vectors of the next words
- *Gensim* model is like a *Python* dictionary
- *fingerprint* is linked to the same memory location as the word embedding in the model!
- They essentially become one.
- Thus, word embedding in the model (say, '*today*') is summed up with the next vectors.

Discussion of the obligatory assignment

Interesting issue with initialization, leading to *Infs*

- After some time, the same word occurs in the text.
- Its vector is added to itself and is doubled!
- *fingerprint* values grow fast and quickly reach *Inf*;
- the model in RAM is corrupted;
- things go crazy.

Remedy:
fingerprint = numpy.zeros(model.vector_size)
fingerprint += model[first_word]
fingerprint += model[second_word]
...
fingerprint += model[last_word]

Discussion of the obligatory assignment

Do we need averaging step at all?

- Only one student tried to use simple sum of word vectors instead of average.
- Classifier performance jumped from 0.68 to 0.75...
- ...with less computation time.
- Why so?

Discussion of the obligatory assignment

Average text length (in words)

- The Daily Mail 389
- 4Traders 327
- Individual.com 229
- Latest Nigerian News 97

Discussion of the obligatory assignment

Capturing non-semantic signals

- Classes differ in typical document length.
- Longer documents produce semantic fingerprints with larger magnitudes (values).
- Averaging normalizes the magnitudes by the number of words: eliminates length differences.
- Without averaging, document vectors remain different.
- Logistic regression happily employs this signal for classification...
- ...but it is not related to document semantics.

Can be considered a sort of overfitting: performance will severely drop if typical text length changes.
Still, a very interesting finding!

Contents

The exam: what to expect?

Nothing extremely difficult at the exam

- Mostly simply answering questions
- ...related to general understanding of the basic concepts
- ...and to practical aspects of prediction-based distributional models.
- At most one problem requiring (simple) calculation.
- The only formula you have to remember by heart is cosine distance.

The exam: what to expect?

Most essential reading

1. Chapters from 'Speech and Language Processing' by Jurafsky and Martin
2. 'From Frequency to Meaning: Vector Space Models of Semantics' by Turney and Pantel
3. 'Word2vec parameter learning explained' by Rong (at least skim through)
4. 'Distributed representations of words and phrases and their compositionality' by Mikolov et al.
5. 'Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change' by Hamilton et al.

The links are at the Syllabus page.

The exam: what to expect?

Exam-like problems at Dec 1 group session

1. Draw the scheme of how CBOW and Continuous Skipgram algorithms train.
2. Briefly describe all key elements of the neural network in these algorithms.
3. Enumerate and briefly describe all ways of standardized extrinsic evaluation of word embedding models that you can think of.
4. How evaluation metrics are related to syntagmatic or paradigmatic relations between words?
5. How many values (parameters) a trained prediction-based model contain?
6. How to estimate its size (in MBytes), if all the values are 32-bit floats?
7. etc...

The exam: what to expect?

Questions?

INF5820
Distributional Semantics: Extracting Meaning from Data
Thanks for your attention!
Good luck at the exam!

References I

- Lazaridou, A., Bruni, E., and Baroni, M. (2014).
  Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world.
  In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1403–1414.
- Nguyen, A. K., Schulte im Walde, S., and Vu, T. N. (2016).
  Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction.
  In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459. Association for Computational Linguistics.

References II

- Parasca, I.-E., Rauter, A. L., Roper, J., Rusinov, A., and Stenetorp, G. B. S. R. P. (2016).
  Defining words with words: Beyond the distributional hypothesis.
  In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, pages 122–126. Association for Computational Linguistics.
- Shwartz, V., Goldberg, Y., and Dagan, I. (2016).
  Improving hypernymy detection with an integrated path-based and distributional method.
  arXiv preprint arXiv:1603.06076.

References III

- Upadhyay, S., Faruqui, M., Dyer, C., and Roth, D. (2016).
  Cross-lingual models of word embeddings: An empirical comparison.
  In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1670. Association for Computational Linguistics.