INF5820
Distributional Semantics: Extracting Meaning from Data

Lecture 3
Practical aspects of training and using distributional models

Andrey Kutuzov
andreku@ifi.uio.no

9 November 2016

Contents

Brief recap

What we are going to cover today

- Models evaluation;
- Off-the-shelf tools to train and use models;
- Models' formats;
- Models hyperparameters.

Contents

Models evaluation

How do we evaluate trained models? Subject to many discussions!

The topic of a special workshop at ACL2016:

`https://sites.google.com/site/repevalacl16/`

Models evaluation

How do we evaluate trained models? Subject to many discussions!
The topic of a special workshop at ACL2016:
`https://sites.google.com/site/repevalacl16/`

- Semantic relatedness (what is the association degree?):
  - RG dataset [Rubenstein and Goodenough, 1965]

Models evaluation

How do we evaluate trained models? Subject to many discussions!
The topic of a special workshop at ACL2016:
`https://sites.google.com/site/repevalacl16/`

- Semantic relatedness (what is the association degree?):
    - RG dataset [Rubenstein and Goodenough, 1965]
    - WordSim 353 dataset [Finkelstein et al., 2001]

Models evaluation

How do we evaluate trained models? Subject to many discussions!
The topic of a special workshop at ACL2016:
`https://sites.google.com/site/repevalacl16/`

- Semantic relatedness (what is the association degree?):
  - RG dataset [Rubenstein and Goodenough, 1965]
  - WordSim 353 dataset [Finkelstein et al., 2001]
  - MEN dataset [Bruni et al., 2014]

Models evaluation

How do we evaluate trained models? Subject to many discussions!
The topic of a special workshop at ACL2016:
`https://sites.google.com/site/repevalacl16/`

- Semantic relatedness (what is the association degree?):
  - RG dataset [Rubenstein and Goodenough, 1965]
  - WordSim 353 dataset [Finkelstein et al., 2001]
  - MEN dataset [Bruni et al., 2014]
  - SimLex-999 dataset [Hill et al., 2015]

Models evaluation

How do we evaluate trained models? Subject to many discussions!
The topic of a special workshop at ACL2016:
`https://sites.google.com/site/repevalacl16/`

- Semantic relatedness (what is the association degree?):
  - RG dataset [Rubenstein and Goodenough, 1965]
  - WordSim 353 dataset [Finkelstein et al., 2001]
  - MEN dataset [Bruni et al., 2014]
  - SimLex-999 dataset [Hill et al., 2015]
- Synonym detection (what is most similar?):
  - TOEFL dataset (1997)

Models evaluation

- Concept categorization (what groups with what?):
  - ESSLI 2008 dataset
  - Battig dataset (2010)

Models evaluation

- Concept categorization (what groups with what?):
  - ESSLI 2008 dataset
  - Battig dataset (2010)
- Analogical inference (A is to B as C is to ?):
  - Google Analogy dataset [Le and Mikolov, 2014]
  - Many domain-specific datasets inspired by Google Analogy

Models evaluation

- Concept categorization (what groups with what?):
  - ESSLI 2008 dataset
  - Battig dataset (2010)
- Analogical inference (A is to B as C is to ?):
  - Google Analogy dataset [Le and Mikolov, 2014]
  - Many domain-specific datasets inspired by Google Analogy
- Correlation with manually crafted linguistic features:
  - QVEC uses words affiliations with *Wordnet* synsets [Tsvetkov et al., 2015]

Contents

Off-the-shelf tools to train and use models

Main frameworks and toolkits

1. *Dissect* [Dinu et al., 2013] (`http://clic.cimec.unitn.it/composes/toolkit/`);

Off-the-shelf tools to train and use models

Main frameworks and toolkits

1. *Dissect* [Dinu et al., 2013] (`http://clic.cimec.unitn.it/composes/toolkit/`);
2. word2vec original C code [Le and Mikolov, 2014] (`https://word2vec.googlecode.com/svn/trunk/`)

Off-the-shelf tools to train and use models

Main frameworks and toolkits

1. *Dissect* [Dinu et al., 2013] (`http://clic.cimec.unitn.it/composes/toolkit/`);
2. word2vec original C code [Le and Mikolov, 2014] (`https://word2vec.googlecode.com/svn/trunk/`)
3. *Gensim* framework for Python, including word2vec implementations (`http://radimrehurek.com/gensim/`);

Off-the-shelf tools to train and use models

Main frameworks and toolkits

1. *Dissect* [Dinu et al., 2013] (`http://clic.cimec.unitn.it/composes/toolkit/`);
2. word2vec original C code [Le and Mikolov, 2014] (`https://word2vec.googlecode.com/svn/trunk/`)
3. *Gensim* framework for Python, including word2vec implementations (`http://radimrehurek.com/gensim/`);
4. word2vec implementations in *Google's TensorFlow* (`https://www.tensorflow.org/tutorials/word2vec`);

Off-the-shelf tools to train and use models

Main frameworks and toolkits

1. *Dissect* [Dinu et al., 2013] (`http://clic.cimec.unitn.it/composes/toolkit/`);
2. word2vec original C code [Le and Mikolov, 2014] (`https://word2vec.googlecode.com/svn/trunk/`)
3. *Gensim* framework for Python, including word2vec implementations (`http://radimrehurek.com/gensim/`);
4. word2vec implementations in *Google's TensorFlow* (`https://www.tensorflow.org/tutorials/word2vec`);
5. GloVe reference implementation [Pennington et al., 2014] (`http://nlp.stanford.edu/projects/glove/`).

Contents

Model formats

Models can come in several formats:

1. Simple text format: words and sequences of values representing their vectors, one word per line; first line gives information on the number of words in the model and vector size.

Model formats

Models can come in several formats:

1. Simple text format: words and sequences of values representing their vectors, one word per line; first line gives information on the number of words in the model and vector size.
2. The same in the binary form.

Model formats

Models can come in several formats:

1. Simple text format: words and sequences of values representing their vectors, one word per line; first line gives information on the number of words in the model and vector size.
2. The same in the binary form.
3. Gensim binary format: uses *NumPy* matrices saved via Python pickles; stores a lot of additional information (input vectors, training algorithm, word frequency, etc).

Gensim works with all of these formats.

Contents

Hyperparameters influence

Things are complicated

Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

1. CBOW or skip-gram algorithm. Needs further research; SkipGram is generally better (but slower). CBOW seems to be better on small corpora (less than 100 mln tokens).

Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

1. CBOW or skip-gram algorithm. Needs further research; SkipGram is generally better (but slower). CBOW seems to be better on small corpora (less than 100 mln tokens).
2. Vector size: how many distributed semantic features (dimensions) we use to describe a word. The more is not always the better.

Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

1. CBOW or skip-gram algorithm. Needs further research; SkipGram is generally better (but slower). CBOW seems to be better on small corpora (less than 100 mln tokens).
2. Vector size: how many distributed semantic features (dimensions) we use to describe a word. The more is not always the better.
3. Window size: context width and influence of distance. **Topical** (associative) or **functional** (semantic proper) models.

Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

1. CBOW or skip-gram algorithm. Needs further research; SkipGram is generally better (but slower). CBOW seems to be better on small corpora (less than 100 mln tokens).
2. Vector size: how many distributed semantic features (dimensions) we use to describe a word. The more is not always the better.
3. Window size: context width and influence of distance. **Topical** (associative) or **functional** (semantic proper) models.
4. Frequency threshold: useful to get rid of long noisy lexical tail;

Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

1. CBOW or skip-gram algorithm. Needs further research; SkipGram is generally better (but slower). CBOW seems to be better on small corpora (less than 100 mln tokens).
2. Vector size: how many distributed semantic features (dimensions) we use to describe a word. The more is not always the better.
3. Window size: context width and influence of distance. **Topical** (associative) or **functional** (semantic proper) models.
4. Frequency threshold: useful to get rid of long noisy lexical tail;
5. Selection of learning material: hierarchical softmax or negative sampling (used more often);

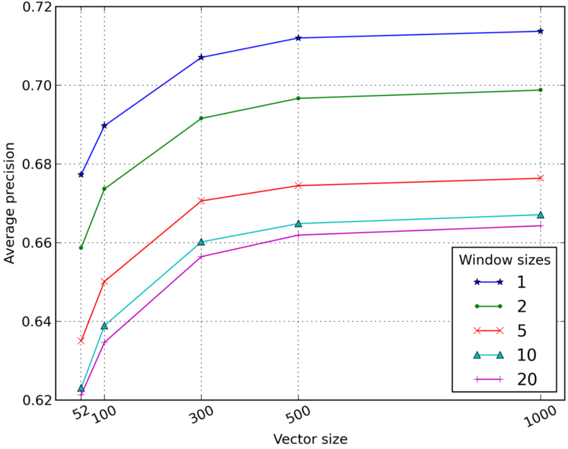
Hyperparameters influence

Things are complicated

Model performance hugely depends on training settings (hyperparameters):

1. CBOW or skip-gram algorithm. Needs further research; SkipGram is generally better (but slower). CBOW seems to be better on small corpora (less than 100 mln tokens).
2. Vector size: how many distributed semantic features (dimensions) we use to describe a word. The more is not always the better.
3. Window size: context width and influence of distance. **Topical** (associative) or **functional** (semantic proper) models.
4. Frequency threshold: useful to get rid of long noisy lexical tail;
5. Selection of learning material: hierarchical softmax or negative sampling (used more often);
6. Number of iterations on our training data, etc...

Hyperparameters influence



Model performance in semantic relatedness task depending on context width and vector size.

Hyperparameters influence

A bunch of observations

- Wikipedia is not the best training corpus: fluctuates wildly depending on hyperparameters. Perhaps, too specific language.

Hyperparameters influence

A bunch of observations

- Wikipedia is not the best training corpus: fluctuates wildly depending on hyperparameters. Perhaps, too specific language.
- Normalize you data: lowercase, lemmatize, merge multi-word entities.

Hyperparameters influence

A bunch of observations

- Wikipedia is not the best training corpus: fluctuates wildly depending on hyperparameters. Perhaps, too specific language.
- Normalize you data: lowercase, lemmatize, merge multi-word entities.
- It helps to augment words with PoS tags before training (*'boot_NOUN'*, *'boot_VERB'*). As a result, your model becomes aware of morphological ambiguity.

Hyperparameters influence

A bunch of observations

- Wikipedia is not the best training corpus: fluctuates wildly depending on hyperparameters. Perhaps, too specific language.
- Normalize you data: lowercase, lemmatize, merge multi-word entities.
- It helps to augment words with PoS tags before training (*'boot_NOUN'*, *'boot_VERB'*). As a result, your model becomes aware of morphological ambiguity.
- Remove your stop words yourself. Statistical downsampling implemented in *word2vec* algorithms can easily deprive you of valuable text data.

Hyperparameters influence

Questions?

INF5820
Distributional Semantics: Extracting Meaning from Data
Lecture 3
Practical aspects of training and using distributional models

Homework: **obligatory assignment 3.**

Contents

In the next week

Beyond words: distributional representations of texts

- Representing phrases, sentences and documents;
- semantic fingerprints;
- paragraph vector (doc2vec);
- deep inverse regression
- etc.

References I

Bruni, E., Tran, N.-K., and Baroni, M. (2014).
Multimodal distributional semantics.
J. Artif. Intell. Res.(JAIR), 49(1-47).

Dinu, G., Pham, T. N., and Baroni, M. (2013).
Dissect - distributional semantics composition toolkit.
In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36. Association for Computational Linguistics.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001).
Placing search in context: The concept revisited.
In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

References II

- Hill, F., Reichart, R., and Korhonen, A. (2015).
  Simlex-999: Evaluating semantic models with (genuine) similarity estimation.
  Computational Linguistics, 41(4).

- Le, Q. V. and Mikolov, T. (2014).
  Distributed representations of sentences and documents.
  In *ICML*, volume 14, pages 1188–1196.

- Pennington, J., Socher, R., and Manning, C. D. (2014).
  GloVe: Global vectors for word representation.
  In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

- Rubenstein, H. and Goodenough, J. B. (1965).
  Contextual correlates of synonymy.
  Communications of the ACM, 8(10):627–633.

References III

Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G., and Dyer, C. (2015).
Evaluation of word vector representations by subspace alignment.
In *Proc. of EMNLP*.