Constructing and Evaluating Word Embeddings

Dr Marek Rei and Dr Ekaterina Kochmar
Computer Laboratory
University of Cambridge

In this topic you have:

- looked into construction of word representations using both traditional (count-based) and various neural network models
- seen that representing words as low-dimensional vectors allows systems to take advantage of semantic similarities, generalise to unseen examples and improve pattern detection accuracy
- looked at a range of tasks (e.g., word similarity, semantic and syntactic analogy)
- learned about recent advances: e.g., multilingual embeddings and multimodal vectors

Count-based models

- Count-based vectors built using word co-occurrence with specific other words
- Baroni et al. (2014). *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors* showed that on most of the tasks predicting models outperform count-based models

Count-based vectors

One way of creating a vector for a word:
Let's count how often a word occurs together with specific other words.

He is reading a **magazine**

I was reading a **newspaper**

This **magazine** published my story

The **newspaper** published an article

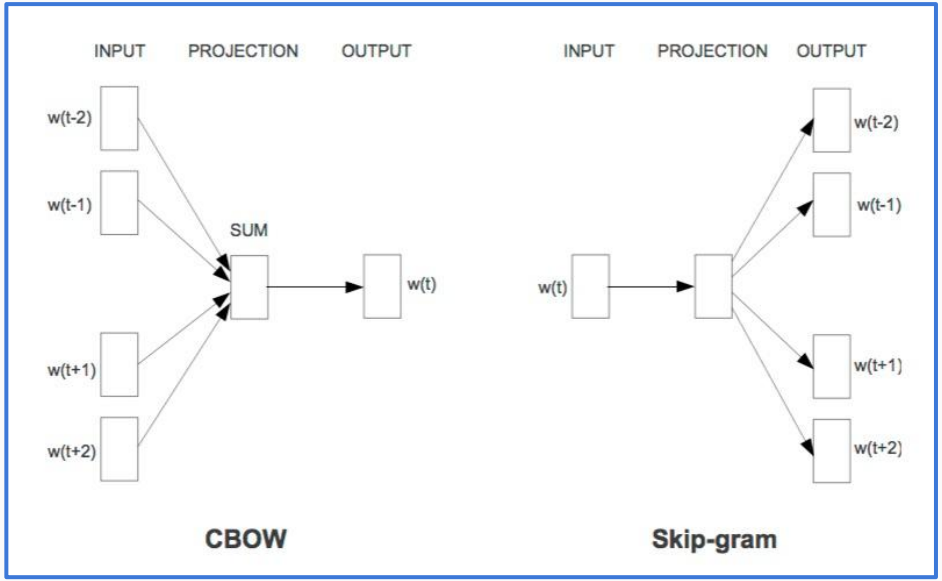
She buys a **magazine** every month

He buys this **newspaper** every day

	reading	a	this	published	my	buys	the	an	every	month	day
magazine	1	2	1	1	1	1	0	0	1	1	0
newspaper	1	1	1	1	0	1	1	1	1	0	1

Predicting models

- Predicting models – predict the current word given the context (**CBOW**), or the surrounding words given the current word (**Skip-gram**)
- Mikolov et al. (2013). *Efficient Estimation of Word Representations in Vector Space*
- Mikolov et al. (2013). *Linguistic Regularities in Continuous Space Word Representations*



Count-based vs. predicting models

- Baroni et al. (2014) showed that predicting models outperform count-based models, *while*
- Levy & Goldberg (2014). *Linguistic Regularities in Sparse and Explicit Word Representations Comparison on different vector types on the linguistic regularities task* showed that relational similarities can be recovered from traditional distributional word representations
- The neural embedding process is not discovering novel patterns, but rather is preserving the patterns inherent in the word-context co-occurrence matrix

Count-based or predicting models?

- Levy & Goldberg (2014). *Linguistic Regularities in Sparse and Explicit Word Representations Comparison on different vector types on the linguistic regularities task* further show that some relations are better captured by count-based models, and some – by neural embeddings
- Ultimately, which type of the models is better?

Essay topic suggestions (I)

- Comparison of the two types of models:
  - Based on the evidence presented in the papers, discuss which type of the models is more suitable for a particular task and why
  - Are the neural embeddings superior to the traditional (count-based) models?
  - Are the two types of models complementary to each other?
- Additional reading:
  - Christopher D. Manning (2015). *Last Words. Computational Linguistics and Deep Learning* (http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00239)
  - Levy et al. (2015). *Improving Distributional Similarity with Lessons Learned from Word Embeddings*

Essay topic suggestions (II)

- Critical evaluation of the approaches overviewed in the course
- In-depth discussion of the approaches presented in the papers and the tasks
- Suggestions for improvements

For example:

- A number of approaches attempted to capture semantics of linguistic units beyond words or even sentences. For example, Moritz Hermann & Blunsom (2014) build document representations using the average of the representations of all document sentences. Discuss how semantics of longer linguistic units can be represented using word embeddings.

Summary of the papers

- We've looked into a number of neural network architectures, involving:
  - use of syntactic relations between words (Socher et al., 2012) vs ignoring the relations (Moritz Hermann & Blunsom, 2014)
  - integration of semantic lexicons (Faruqui et al., 2015)
- Multilingual embeddings (Moritz Hermann & Blunsom, 2014)
- Multimodal vectors (Norouzi et al., 2014)
- A range of tasks using a number of different datasets

Overview of the tasks

Semantic relatedness	Baroni et al. (2014), Faruqui et al. (2015)
Synonym detection	Baroni et al. (2014), Faruqui et al. (2015)
Concept categorisation	Baroni et al. (2014)
Selectional preferences	Baroni et al. (2014)
Analogy recovery	Mikolov et al. (2013), Baroni et al. (2014), Levy & Goldberg (2014), Faruqui et al. (2015)

Overview of the tasks

Sentiment analysis	Socher et al. (2012), Faruqui et al. (2015)
Classification of semantic relationships	Socher et al. (2012)
Cross-lingual document classification	Moritz Hermann & Blunsom (2014)
Image labelling	Norouzi et al. (2014)

Project suggestions

- Replicate the experiments reported in one of the papers:
  - using a different dataset (see the datasets on http://www.wordvectors.org/), *or*
  - using a different architecture (e.g., see different pre-trained vectors http://www.marekrei.com/projects/vectorsets/), *or*
  - using the same setting on a different task, *or*
  - introducing small (= doable within the time limit allowed for the projects) improvements to the models
- and report the results comparing them to the previous work

Datasets & Resources

- **Word2vec** – a tool for creating word embeddings: https://code.google.com/archive/p/word2vec/
- Word vectors **pretrained** on 100B words. More information on the word2vec homepage: https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing
- An online tool for evaluating word vectors on 12 different word similarity datasets with the **links to the datasets**: http://www.wordvectors.org/

Datasets & Resources

- Tool for **converting** word2vec vectors between binary and plain-text formats. You can use this to convert the pre-trained vectors to plain-text: https://github.com/marekrei/convertvec
- Vectors trained using **3 different methods** (counting, word2vec and dependecy-relations) on the same dataset (BNC): http://www.marekrei.com/projects/vectorsets/
- **t-SNE**, a tool for visualising word embeddings in 2D: http://lvdmaaten.github.io/tsne/

Datasets & Resources

- **Retrofitting** word vectors to semantic lexicons (Faruqui et al., 2015): https://github.com/mfaruqui/retrofitting
- **GloVe**: Global vectors for word representation (Pennington et al., 2014): http://nlp.stanford.edu/projects/glove/
- **Global context** vectors (Huang et al., (2012). *Improving Word Representations via Global Context and Multiple Word Prototypes*): http://www.socher.org/index.php/Main/ImprovingWordRepresentationsViaGlobalContextAndMultipleWordPrototypes
- **Multilingual** vectors (Faruqui & Dyer, (2014). *Improving Vector Space Word Representations Using Multilingual Correlation*): http://www.cs.cmu.edu/~mfaruqui/soft.html

Further reading

- Lecture slides: http://www.cl.cam.ac.uk/teaching/1516/R222/materials.html
- Multimodal vectors:
  - Kiros et al. (2015). *Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models*
  - Frome et al. (2013). *DeViSE: A Deep Visual-Semantic Embedding Model*
- More on the use of the NN models:
  - Socher et al. (2012). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*
  - Socher et al. (2011). *Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection*
  - Bengio et al. (2003). *A Neural Probabilistic Language Model*

Assessment

- Undertake a small project and write an associated report (5000 words), *or*
- Write an essay addressing a research issue (5000 words)
- Email us with any further questions and to agree on the project/essay topic

Next topics

- **Integrating Compositional and Distributional Semantics** – related to the traditional (count-based) models overviewed in this topic
- **Applications of Neural Networks** – on further use of the NNs for linguistic tasks such as tagging, parsing, machine translation, among others