Word embeddings, information retrieval and textual entailment
—

Eric Gaussier
(Abdelkader El Mahdaouy, James Henderson, Said Ouatik El Alaoui, Julien Perez, Diana Popa)

Univ. grenoble Alpes, CNRS, Grenoble INP - LIG / AMA

26 March 2018

Table of contents

to embed

1. To lay as in a bed; to lay in surrounding matter; to bed.

   to embed something in clay, mortar, or sand

   ...

4. To define a one-to-one function from (one set) to another so that certain properties of the domain are preserved when considering the image as a subset of the codomain.

   The torus can be embedded in $\mathbb{R}^3$

From *en.wiktionary.org/wiki/embed*

Summary

IR, Textual Entailment, Word Embeddings

IR and textual entailment are similarity based problems!

- IR: From queries to relevant documents through text collections
- Textual entailment: Does sentence 1 entails sentence 2?

IR, Textual Entailment, Word Embeddings

IR and textual entailment are similarity based problems!

- IR: From queries to relevant documents through text collections
- Textual entailment: Does sentence 1 entails sentence 2?

Sentence 1 *A soccer game with multiple males playing.*
Sentence 2 *Some men are playing a sport.*

IR, Textual Entailment, Word Embeddings

IR and textual entailment are similarity based problems!

- IR: From queries to relevant documents through text collections
- Textual entailment: Does sentence 1 entails sentence 2?

Sentence 1 *A soccer game with multiple males playing.*
Sentence 2 *Some men are playing a sport.*

Sentence 1 *A smiling costumed woman is holding an umbrella.*
Sentence 2 *A happy woman in a fairy costume holds an umbrella.*

Both tasks can be (are) addressed by designing appropriate similarities

IR, Textual Entailment, Word Embeddings

IR and textual entailment are similarity based problems!

- IR: From queries to relevant documents through text collections
- Textual entailment: Does sentence 1 entails sentence 2?

Sentence 1 *A soccer game with multiple males playing.*
Sentence 2 *Some men are playing a sport.*

Sentence 1 *A smiling costumed woman is holding an umbrella.*
Sentence 2 *A happy woman in a fairy costume holds an umbrella.*

Both tasks can be (are) addressed by designing appropriate similarities

(soccer game/sport), (multiple males/men), (play/.)

IR, Textual Entailment, Word Embeddings

IR and textual entailment are similarity based problems!

- IR: From queries to relevant documents through text collections
- Textual entailment: Does sentence 1 entails sentence 2?

Sentence 1 *A soccer game with multiple males playing.*
Sentence 2 *Some men are playing a sport.*

Sentence 1 *A smiling costumed woman is holding an umbrella.*
Sentence 2 *A happy woman in a fairy costume holds an umbrella.*

Both tasks can be (are) addressed by designing appropriate similarities

(soccer game/sport), (multiple males/men), (play/.)

(smiling/happy), (costume/.), (woman/.), (hold/.), (umbrella/.) - fairy?

IR, Textual Entailment, Word Embeddings (cont.)

Similarities

- One needs similarities between documents and sentences
- Based on appropriate word representations

Word representations

The word representations should capture latent (hidden topical, semantic) properties of words
$\rightarrow$ word embeddings

Word embeddings

It did not start in 2013!

Latent Semantic Analysis/Indexing (matrix factorization) - Deerwester et al., 1990

- Let X denote the word-document matrix
- Singular value decomposition of X: $X = U\Sigma V$, where U and V are orthonormal matrices and Σ diagonal
- U_i (V_j) (unweighted) representations for word i (document j)
- Trunk the decomposition to first k dimensions: $X \sim U_k \Sigma_k V_k$ to obtain a *latent concept space* for words and documents
- Comparison is done in this new space ($d_j^c = \Sigma_k V_j$)

What does one capture in the latent concept space?

Word similarities based on co-occurrence properties at the document level (two words co-occurring often will have similar representations)

Topical, and partially, semantic dimensions

Word embeddings (cont.)

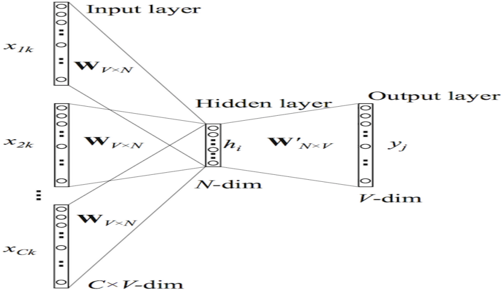
Probabilistic Latent Semantic Analysis/Indexing - T. Hofmann, 1999

- Probabilistic counterpart: $P(w|d) = \sum_{z=1}^{K} P(w|z)P(z|d)$
- Distribution $P(z|w)$ latent representation for w
- Both $P(w|d)$ and $P(z|d)$ can be used as representations for d for retrieval
- Evolution towards LDA (Latent Dirichlet Allocation), Blei et al. 2003 & Fisher Kernels, Nyffenegger et al., 2006

What does one capture in the latent space?
Word similarities based on co-occurrence properties at the document level (two words co-occurring often will have similar representations)
Topical, and partially, semantic dimensions

Word embeddings (cont.)

CBOW (Mikolov et al., 2013)

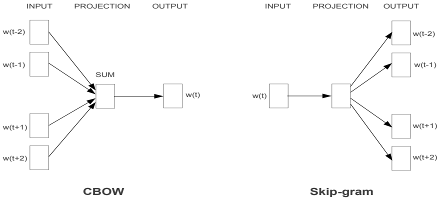


- Objective: maximize $\sum_{t=1}^{|\mathcal{V}|} \log(P(w_t|w_{t-c}, ..., w_{t+c}))$
- 1-hot encoding for input/output
- j^{th} column of input matrix W new representation for words

Image from A. Minnaar's tutorial, 2015

Word embeddings (cont.)

Skip-gram (Mikolov et al., 2013)



- Objective: maximize $\sum_{t=1}^{|\mathcal{V}|} \sum_{j=t-c}^{t+c} \log(P(w_j|w_t))$
- $P(w_o|w_i) = \frac{exp({v'_{w\,o}}^T v_{w\,i})}{\sum_{w \in \mathcal{V}} exp({v'_w}^T v_{w\,i})}$
- 1-hot encoding for input/output
- v_w new representation for words

Image from Mikolov et al., 2013

Word embeddings (cont.)

CBOW, Skip-gram (Mikolov et al., 2013)

What does one capture in the latent space?

Word similarities based on the probability of predicting the words in a local (restricted) context

Semantic, and partially, topical dimensions

Word embeddings (cont.)

Glove (Pennington et al., 2014)

A matrix factorization method based on local contexts:

- X co-occurrence matrix: X_{ij} = numb. of times word j occurs in context of word i
- Objective: minimize $\sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} \text{weight}(X_{ij}) \left(w_i^T w_j + b_i + b_j - \log(X_{ij})\right)^2$
- $\text{weight}(X_{ij}) = (\frac{X_{ij}}{x_{max}})^{\alpha}$
- $x_{max} = 100, \ \alpha = 3/4$
- w is the word embedding

What does one capture in the latent space?

Word similarities based on the co-occurrences of the words in a local (restricted) context
Semantic, and partially, topical dimensions

First conclusions

- A not so recent notion, used in different fields (IR, NLP) from late 80s, early 90s
- A recent focus on local contexts to better capture semantic dimensions
- Strong relations with (neural) language models for CBOW and skip-grams

Summary

Ad hoc IR

Reference paper: Zhang et al., 2016

1. Word embeddings are used to define similarities between query/document words
    - Mostly based on language models and the noisy channel translation model (Berger and Lafferty, 1999)
    - $P(w_q|w_d)$ estimated on the basis of cosine between word embeddings (Ganguly et al., 2015; Zuccon et al, 2015)
2. Word embeddings are used to construct query/document representations (Zheng and Callan, 2015; Zamani and Croft, 2016)
    - Standard vector-based similarities can then be used (Vulic and Moens, 2015)
    - Dual embedding space (input for queries, output for documents) in Naslinick et al., 2016 and Mitra et al., 2016
    - Local embeddings (based on retrieved documents), Diaz et al., 2016
3. Exploiting word embedding for query expansion & PRF
    - Word embeddings are used to select similar terms, in the embedding space, using similarity scores of kNN approaches (Al Masri et al., 2016; Roy et al., 2016; Rekabsaz et al., 2016)
    - Similar approach is used in El Mahdaouy et al., 2018 (to appear)

A generic approach

Integrating word embedding similarities in state-of-the-art IR models (El Mahdaouy et al., 2018)
Most state-of-the-art IR models take the following form (Clinchant and Gaussier, 2010):

$$RSV(q,d) = \sum_{w \in q \cap d} A(w,q) B(w,d,C)$$

with $A(w,q)$ weight of w in query q ($\frac{x_w^q}{l_q}$) and $B(w,d,\mathcal{C})$ weight of w in doc/collection.
For:

- BM25: $B(w,d,\mathcal{C}) = \frac{(k_1+1)x_w^d}{K+x_w^d} \log \frac{N-N_w+0.5}{N_w+0.5}$
- Dirichlet language model: $B(w,d,\mathcal{C}) = \log \frac{x_w^d + \mu \frac{x_w^C}{|C|}}{l_d+\mu}$
- Log-logistic model: $B(w,d,\mathcal{C}) = \log \frac{N_w}{N_w + N t_w^d}$ with $t_w^d = x_w^d \log(1 + c \frac{l_d}{l_{avg}})$

A generic approach (cont.)

Let:
$S_d(w) = Topk(\{w' \in d, cos(w, w') \geq \theta_s\})$ (k most "similar" words to w in d)
and:

$$\mathcal{A}(w, w', d) = \lambda_d \frac{cos(w, w')}{\sum_{w'' \in d} cos(w, w'')}, \text{ if } w \neq w' \text{ (1 otherwise)}$$

One can refine the RSV score through (Li and Gaussier, 2012):

$$RSV(q, d) = \sum_{w \in q} A(w, q) \sum_{w' \in S_d(w)} \mathcal{A}(w, w', d) B(w', d, C)$$

$$(RSV(q, d) = \sum_{w \in q \cap d} A(w, q) B(w, d, C))$$

A generic approach (cont.)

A detour through heuristic retrieval constraints (Fang and Zhai, 2006)

Constraint 1 *Let* $q = w_q$ *be a single-word query and* $d_1 = w_1$ *and* $d_2 = w_2$ *be two single-word documents. If* $s(w_q, w_1) \geq s(w_q, w_2)$*, then* $RSV(q, d_1) \geq RSV(q, d_2)$*.*

Constraint 2 *Let* $q = w_q$ *be a single-word query and* w *be a non-query word such that* $s(w_q, w) > 0$*. If* d_1 *and* d_2 *are two documents such that* $l_{d_1} = 1$*,* $x_{w_q}^{d_1} = 1$*,* $l_{d_2} = k$*,* $x_w^{d_2} = k$ *(*$k \geq 1$*), then* $RSV(q, d_1) > RSV(q, d_2)$*.*

Constraint 2 *Let* $q = \{w_1, w_2\}$ *be a query with only two equally important query words and* w_3 *be a non-query word such that* $s(w_3, w_2) > 0$*. Let* d_1 *and* d_2 *be two documents. If* $l_{d_1} = l_{d_2} > 1$*,* $x_{w_1}^{d_1} = l_{d_1}$*,* $x_{w_1}^{d_2} = l_{d_2} - 1$*,* $x_{w_3}^{d_2} = 1$*, then* $RSV(q, d_1) < RSV(q, d_2)$*.*

Property *The first constraint is satisfied for all models; the second and third constraints provide upper and lower bounds on the possible value of* λ_d *(learned through cross-validation)*

A generic approach (cont.)

Illustration on Arabic IR (TREC 2001/2002, Farasa stemmer (Abdelali et al., 2016))

	Baseline		$\mathcal{S}_d$					
			CBOW		SKIP-gram		Glove	
Model	MAP	P10	MAP	P10	MAP	P10	MAP	P10
LGD	32.42	47.33	34.63^b	49.60	34.36^b	47.87	34.15^b	49.20
SPL	33.51	50.67	36.34^b	51.60	36.2^b	51.47	36.41^b	52.30
BM25	33.42	49.60	35.47^b	51.20	35.38^b	51.60	35.59^b	52.00
LM	31.15	46.39	33.65^b	48.53	33.51^b	47.60	33.47^b	50.00

Conclusions

- Word embeddings used to define word-word similarities
- Yield state-of-the-art retrieval models
- Input representation for end-to-end neural models (dynamic IR)

Summary

Particularities of textual entailment

A tricky entailment example

The dog chased the cat $\Longrightarrow$ *The cat is frightened*

Particularities of textual entailment

A tricky entailment example

The dog chased the cat $\Longrightarrow$ *The cat is frightened*

The dog chased the cat. $\not\Longrightarrow$ *The cat is frightened.*

Syntax-aware token embeddings

1. Aim: find embeddings that capture both semantic and syntactic information
   S1: the dog chased the cat
   S2: the cat chased the mouse
   In *S1*, cat is a semantic patien; in *S2* it is a semantic agent.
2. Differs from semantic disambiguation
3. Idea: start with traditional word embeddings and specialize them through syntactic relations
4. Use tensor factorization for specialization (Trouillon et al., 2016)

Syntax-aware token embeddings (cont.)

Once sentences have been parsed, overall loss:

$$\min_{s \in \mathcal{S}} \alpha T_{loss}^{s} + (1 - \alpha) R_{l}^{s} oss$$

with: $T_{loss}^{s} = \sum \max(0, \gamma + < e_{i'}^{s}, R_{k'}, e_{j'}^{s} > - < e_{i'}^{s}, R_{k'}, e_{j'}^{s} >$

and: $R_{l}^{s} oss = \sum_{e_i^s \in s} - \log(\sigma(e_i^s . w(e_i^s)))$ ($w(e_i^s)$: word embedding of e_i^s)

Yields simple adaptation that can be learned through standard gradient descent techniques

Can be used as input to CNN/LSTM for sentence similarity or textual entailment computation

Illustration on sentence classification

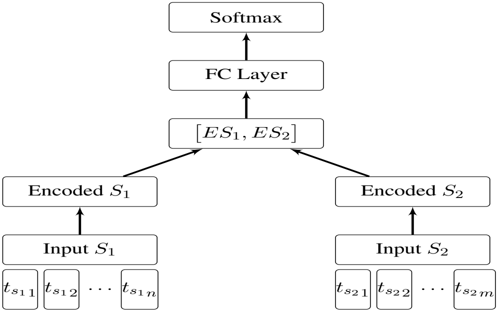


Illustration (cont.)

	Model	MSRPC	SICK	SNLI-20k
Glove	LSTM	0.6863	0.6196	0.5713
	CNN	0.6689	0.6023	0.5584
Glove + positional encoding	LSTM	0.695	0.6135	0.5574
	CNN	0.6718	0.5978	0.5512
Glove + self-attention	LSTM	0.6817	0.5876	0.5062
	CNN	0.6852	0.6174	0.5648
Our	LSTM	0.6927	**0.6359**	0.5676
	CNN	**0.7032**	0.6253	**0.5772**
Our + positional encoding	LSTM	0.6863	0.6153	0.529
	CNN	0.6886	0.6188	0.553
Our + self-attention	LSTM	0.6968	0.6202	0.4952
	CNN	0.6979	0.5686	0.5609

Sentence pair classification accuracy results (2 classes for MSPRC, 3 for SICK and SNLI

Conclusion

1. Widely used representations (NLP, IR)
2. Embedding depends on the information one wants to capture
3. Understanding what is captured here (syntax?, semantic dimensions?)

Bibliography

1. David M. Blei, Andrew Y. Ng, Michael I. Jordan: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 2003
2. Stephane Clinchant, Eric Gaussier: Information-based models for ad hoc IR. SIGIR, 2010
3. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, Richard A. Harshman: Indexing by Latent Semantic Analysis. JASIS 41(6), 1990
4. Abdelkader El Mahdaouy, Saïd El Alaoui Ouatik, Eric Gaussier: Improving Arabic information retrieval using word embedding similarities. I. J. Speech Technology 21(1), 2018
5. Hui Fang, ChengXiang Zhai: Semantic term matching in axiomatic approaches to information retrieval. SIGIR, 2006
6. Bo Li, Eric Gaussier: An Information-Based Cross-Language Information Retrieval Model. ECIR, 2012
7. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space, CoRR abs/1301.3781, 2013

Bibliography (cont.)

8. Alex Minaar: Word2Vec Tutorial Part II: The Continuous Bag-of-Words Model (mc-cormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_II_The_Co of-Words_Model.pdf), 2015
9. Martin Nyffenegger, Jean-Cedric Chappelier, Eric Gaussier: Revisiting Fisher Kernels for Document Similarities. ECML, 2006
10. Jeffrey Pennington, Richard Socher, Christopher D. Manning: Glove: Global Vectors for Word Representation. EMNLP, 2014
11. Theo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, Guillaume Bouchard: Complex Embeddings for Simple Link Prediction. ICML, 2016
12. Ye Zhang, Md. Mustafizur Rahman, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, Aaron Angert, Edward Banner, Vivek Khetan, Tyler McDonnell, An Thanh Nguyen, Dan Xu, Byron C. Wallace, Matthew Lease: Neural Information Retrieval: A Literature Review. CoRR abs/1611.06792, 2016