Neural Approaches to Analogy

ATA@ICCBR2022

Claire Gardent

What is an analogy ?

Formal View

Geometric proportion, geometric arithmetic proportion and as a parallelogram in a vector space

$$(i) \quad \frac{A}{B} = \frac{C}{D} \quad (ii) \quad A - B = C - D \quad (iii) \quad \overrightarrow{A} - \overrightarrow{B} = \overrightarrow{C} - \overrightarrow{D}$$

Postulates

$\forall a, b, c, d \in X :$

- $a : b :: a : b$ (reflexivity)
- $a : b :: c : d \rightarrow c : d :: a : b$ (symmetry)

What is an analogy ?

Informal View

- Two pairs that have a ***high degree of relational similarity*** are analogous (Turney, 2006).
- Two pairs linked by the same relation
- Analogies are defined as relational similarities between two pairs of entities such that the relation that holds between the entities of the first pair, also holds for the second pair.
- Let a, b, c, d be four values from a domain X. The quadruple (a, b, c, d) is said to be in analogical proportion a : b :: c : d if a is related to b as c is related to d, i.e.,

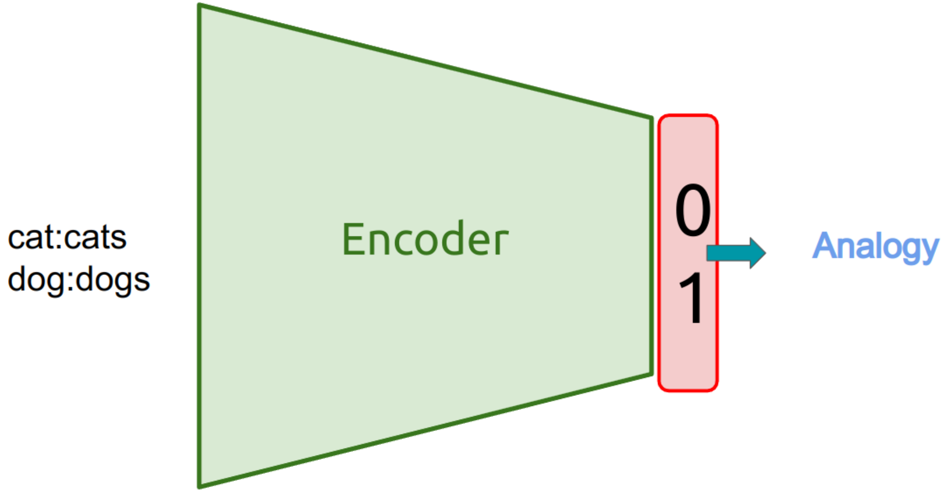
$$R(a, b) \sim R(c, d)$$

Neural Approaches to Analogy

Three main tasks

Classify

$A:B :: C:D \rightarrow \{0,1\}$



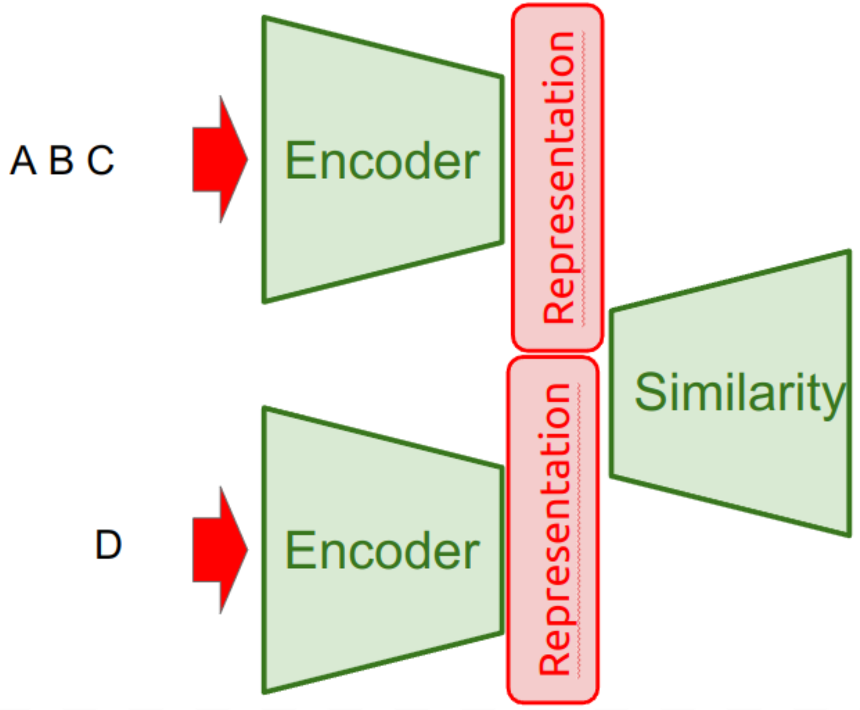
Neural Approaches to Analogy

Three main tasks

Retrieve

A:B :: C $\rightarrow$ D

- Encode A, B and C into $\overrightarrow{ABC}$
- Retrieve D such that $argmax_D\, sim(\overrightarrow{ABC}, \overrightarrow{D})$



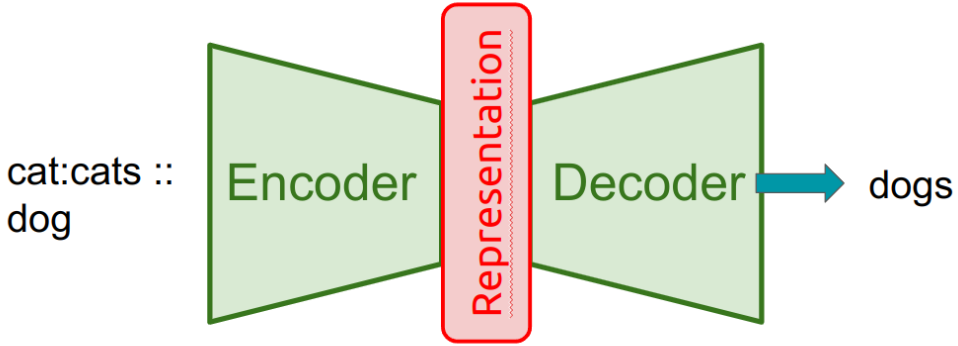
Neural Approaches to Analogy

Three main tasks

Generate

A:B :: C $\rightarrow$ D

- Encode A, B and C into $\overrightarrow{ABC}$
- Decode C from $\overrightarrow{ABC}$



Outline

Image Captions

- Using analogies between image captions to handle ***unseen queries***

Lexical analogies

- Using analogies to ***evaluate word embeddings***
- Learning word embeddings which capture analogies

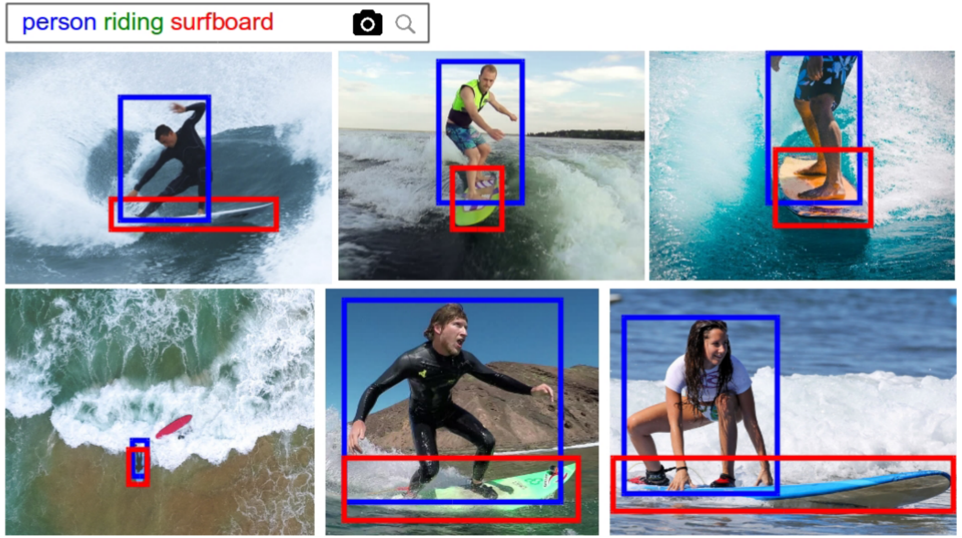
Sentential Analogies

- Using analogies to ***improve retrieval based QA***
- Using analogies to ***evaluate sentence embeddings***
- Learning sentence embeddings which capture analogies

Cross-Modal Text/Image Retrieval

Task : Image Retrieval

Given a language query such as (person, riding, surfboard), retrieve an image which satisfies that query



Peyre et al. 2019

Challenge and Analogical Reasoning

Not all (s,p,o) combinations are seen at training time

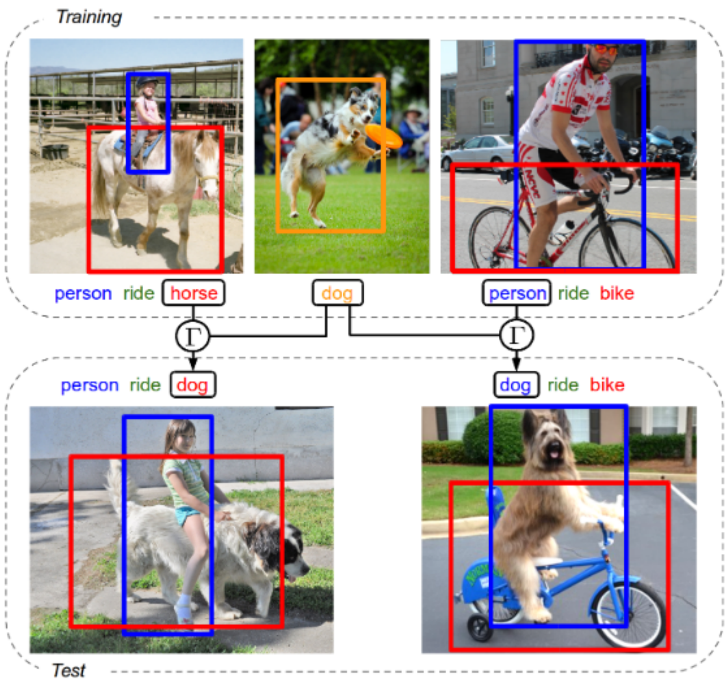
How to handle ***unseen queries***?

Analogical transfer is used to create query representations for unseen triples which are close to the corresponding images

Use analogies to handle unseen queries

Image Retrieval by Analogy

person ride dog = person ride horse - horse + dog



- Unseen triple *"person ride dog"*
- Retrieve neighbours
    - person ride horse
    - person ride bike
    - dog ride bike
- Apply ***analogical transfer*** to neighbours and aggregate
- Retrieve image whose embedding is closest to aggregated embedding

Input

Image

- Bounding box for subject and object
  $i = (s, o)$
- Text/Image: Vector specifying which p relation holds in i
  $y_t^i = 1$ if relation p holds in i
  else $y_t^i = 0$

Text

- Triplet describing the image
  $t = (s', p, o')$

Visual embeddings

- Subject, Object: From CNN pretrained on object detection
- Relation: 8-dimensional vector that concatenates the subject and object box coordinates renormalized with respect to the union box
- Visual phrase: Concatenation of projections of s, p and o

Language embeddings

- Word2Vec embeddings

Common Embedding Space

Two mappings are learned to map text and image embedding into a common space

- For each type of visual embedding (b = subject, object, relation or visual phrase)

$$v_i^b = f_v^b(x_i)$$

- For each type of language embedding

$$w_t^b = f_w^b(q_t)$$

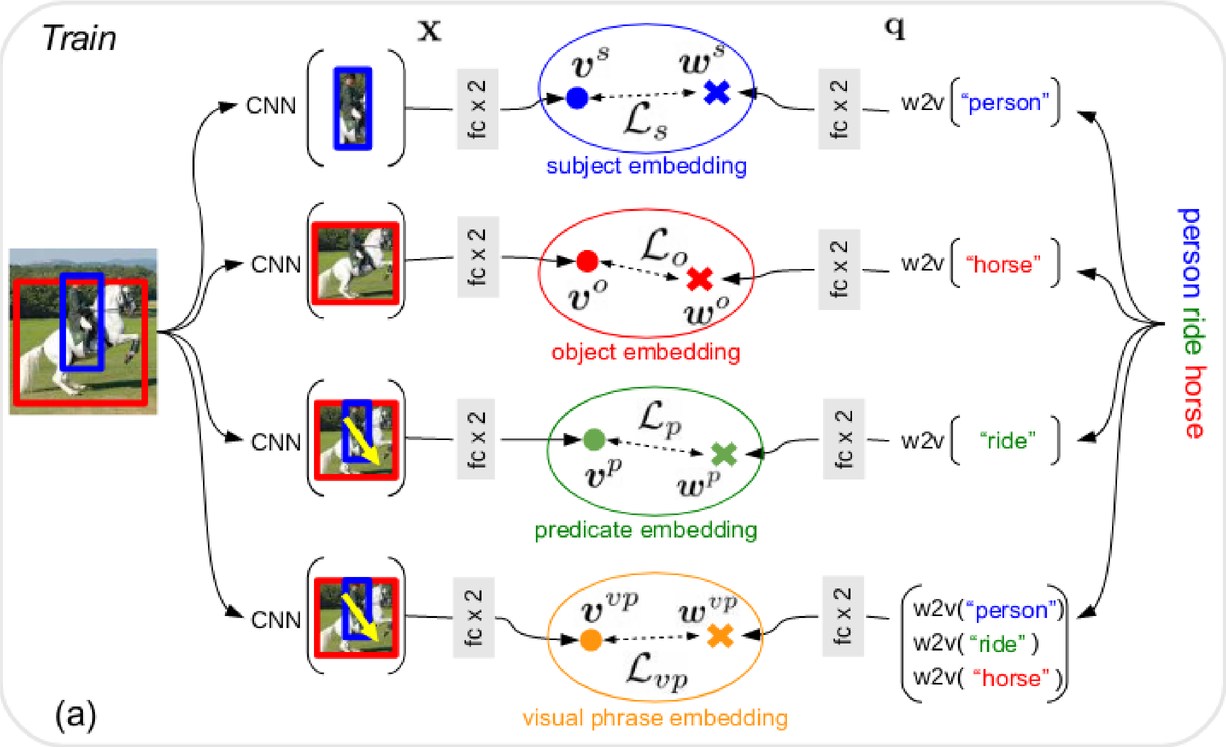
Learned by maximising log likelihood

$$\mathcal{L}_b = \sum_{i=1}^{N} \sum_{t \in \mathcal{V}_b} \mathbb{1}_{y_t^i = 1} \log \left(\frac{1}{1 + e^{-\boldsymbol{w}_t^{b\,T} \boldsymbol{v}_i^b}} \right) + \sum_{i=1}^{N} \sum_{t \in \mathcal{V}_b} \mathbb{1}_{y_t^i = 0} \log \left(\frac{1}{1 + e^{\boldsymbol{w}_t^{b\,T} \boldsymbol{v}_i^b}} \right),$$

Joint loss (One loss per input type)

$$\mathcal{L}_{joint} = \mathcal{L}_s + \mathcal{L}_o + \mathcal{L}_p + \mathcal{L}_{vp}$$

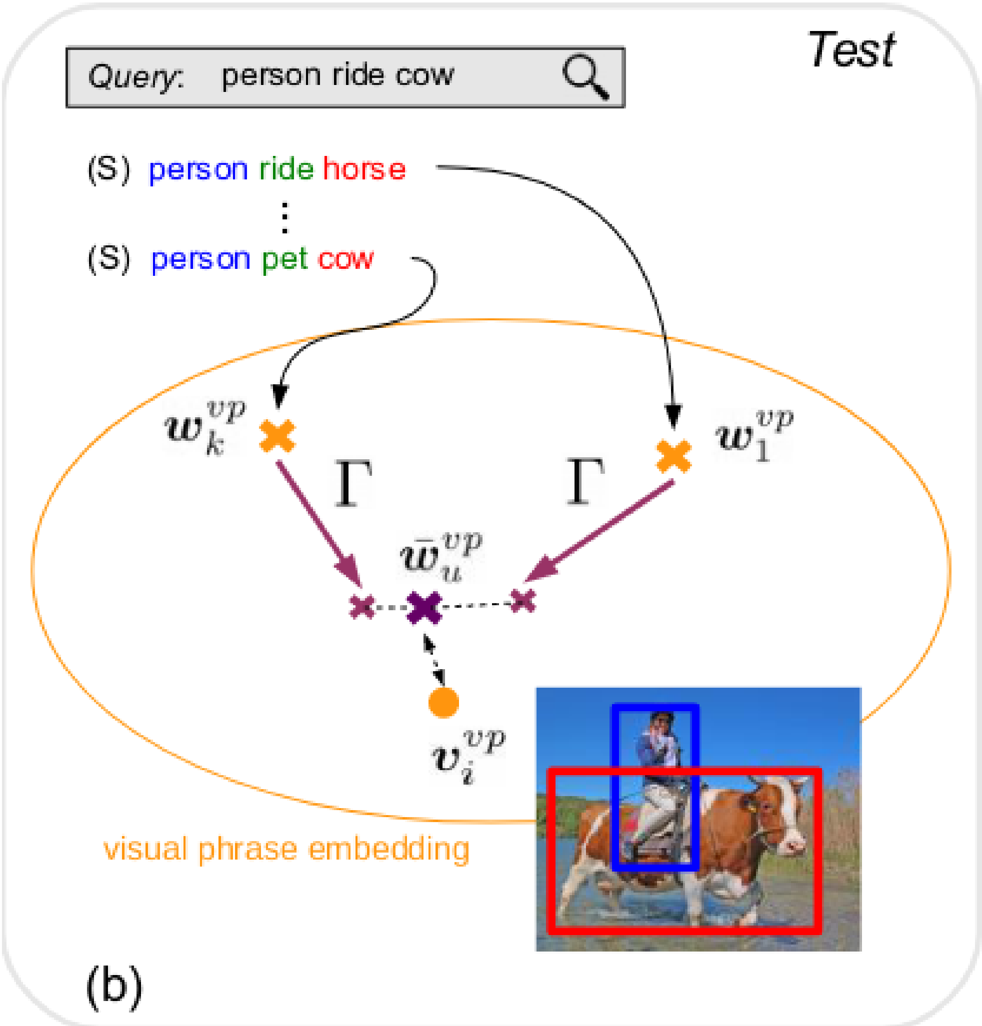
Image and Text Embeddings



Retrieval

To retrieve an image i, given the ***unseen (target) triple*** t'

- retrieve a set of triples $N(t')$ that are similar to t'
- apply analogy transfer to create an embedding $A(t')$ for t' using triples from $N(t')$
- Retrieve image i whose embedding is closest to aggregated transferred embeddings



Retrieving similar triples

"person ride dog"

- person ride horse
- person ride bike
- dog ride bike
- ~~man ride bus~~

Similarity Function for selecting Neighbours

$$G(t, t') = \sum_{b \in s,p,o} \alpha_b \, w_t^b \top w_{t'}^b$$

- Decomposes the similarity between triplets t and t' by looking at the similarities between their subjects, predicates and objects measured by the dot-product of their embeddings.
- α_b weighs the relative contribution of s, p and o to similarity
- Retrieve k most similar triples/images according to G

Analogical Transform

Create embedding for unseen query t' by applying analogical transfer to the embedding of a neighbour

$$\boldsymbol{w}_{t'}^{vp} = \boldsymbol{w}_{t}^{vp} + \Gamma \begin{bmatrix} \boldsymbol{w}_{s'}^{vp} - \boldsymbol{w}_{s}^{vp} \\ \boldsymbol{w}_{p'}^{vp} - \boldsymbol{w}_{p}^{vp} \\ \boldsymbol{w}_{o'}^{vp} - \boldsymbol{w}_{o}^{vp} \end{bmatrix}$$

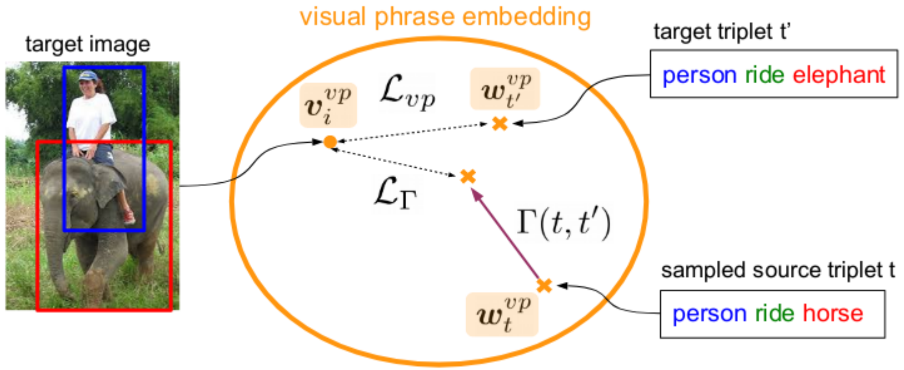
- $t = (s, p, o)$, source seen triplet
- $t' = (s', p', o')$, target unseen triplet
- w_s^{vp}, vp embedding of subject
- Γ = Multi Layer Perceptron

Representing Analogies

$$C = A + \Gamma(C - A)$$

Learning Γ

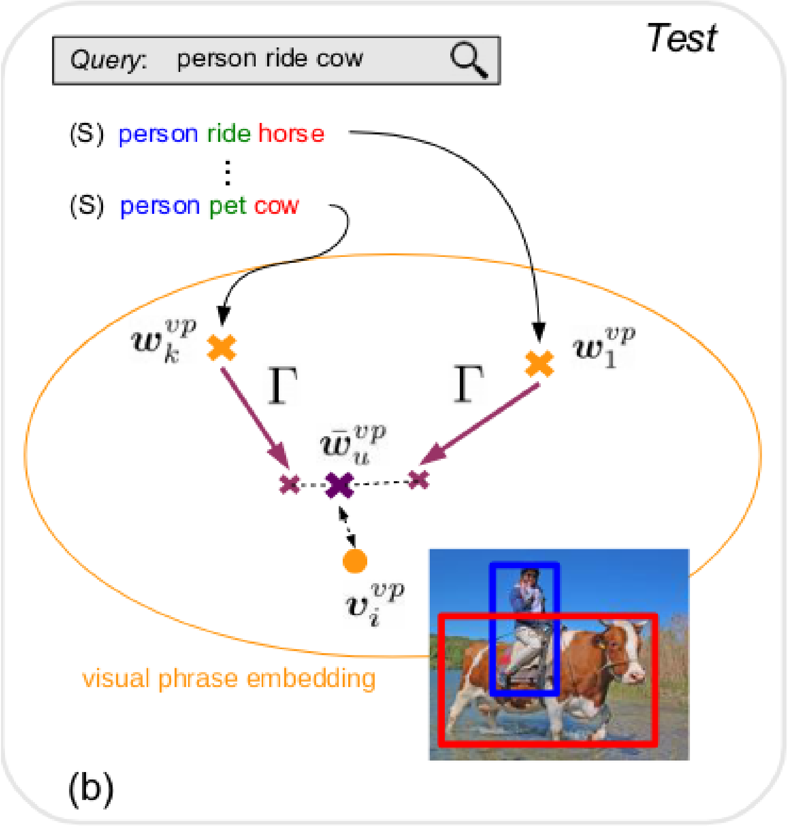
- Maximise log likelihood of the training data
- Trained on negative and positive pairs (t, t')
    - Negative: t' is not similar to t
- Ranking loss pushed the transformed language embedding $w_t^{vp} + \Gamma(t, t')$ to be close to the image embedding v_i^{vp} of the target triplet.



Inference

Given some unseen image description u

- Retrieve the set $N(u)$ of k most similar image descriptions
- Compute and aggregate the Analogical transform for each $t \in N(u)$. This creates an image embedding $\vec{w}_u^{vp}$
- Retrieve the image whose embedding v_i^{vp} is closest to $\vec{w}_u^{vp}$



Results

Comparison with the State of the Art

- Improves the current state-of-the art by more than 30% in terms of relative gain (Table 1)

	full	rare	non-rare
Chao [5]	7.8	5.4	8.5
Gupta [12]	9.1	7.0	9.7
Gkioxari [11]	9.9	7.2	10.8
GPNN [33]	13.1	9.3	14.2
iCAN [9]	14.8	10.5	16.1
s+o+p	18.7	13.8	20.1
s+o+vp	17.7	11.6	19.5
s+o+p+vp	**19.4**	**14.6**	**20.9**

Table 1: Retrieval results on HICO-DET dataset (mAP).

Performance on Unseen Data

- Significantly improves results for unseen triplets (Table 2)

	Base	With aggregation G			
	-	Γ=∅	Γ=0	Γ=$linear$	Γ=$deep$
s+o+p	23.2	-	-	-	-
s+o+vp+transfer	24.1	9.6	24.8	27.6	**28.6**
s+o+p+vp+transfer	23.6	12.5	24.5	25.4	**25.7**
supervised	33.7	-	-	-	-

Table 2: mAP on the 25 zero-shot test triplets of HICO-DET with variants of our model trained on the *trainval* set excluding the positives for the zero-shot triplets. The first column shows the results without analogy transfer (Section 3.1) while the other columns display results with transfer using different forms of analogy transformation Γ (Section 3.2). Last line (supervised) is the

Summary

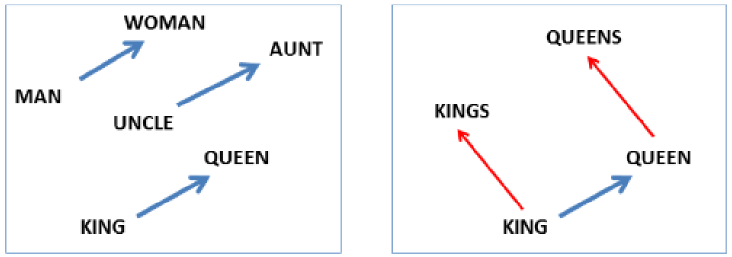
Analogical reasoning to generalise to unseen queries

$$C = A + \Gamma(C - A)$$

Lexical Analogies

Linguistic Regularities in Continuous Space Word Representations

- Neural word embeddings capture syntactic and semantic regularities
- These regularities are observed as ***constant vector offsets*** between pairs of words sharing a particular ***relationship*** .



Singular/Plural Relation

$$\overrightarrow{king} - \overrightarrow{kings} \approx \overrightarrow{queen} - \overrightarrow{queens}$$

Goal: Intrinsic evaluation of word embeddings

Mikolov 2013

Two Analogical Reasoning Tasks

Resolution of Syntactic Lexical Analogies

see:saw :: return:returned

- Test set: 8K instances

Category	Relation	Patterns Tested	# Questions	Example
Adjectives	Base/Comparative	JJ/JJR, JJR/JJ	1000	good:better rough:___
Adjectives	Base/Superlative	JJ/JJS, JJS/JJ	1000	good:best rough:___
Adjectives	Comparative/ Superlative	JJS/JJR, JJR/JJS	1000	better:best rougher:___
Nouns	Singular/Plural	NN/NNS, NNS/NN	1000	year:years law:___
Nouns	Non-possessive/ Possessive	NN/NN_POS, NN_POS/NN	1000	city:city's bank:___
Verbs	Base/Past	VB/VBD, VBD/VB	1000	see:saw return:___
Verbs	Base/3rd Person Singular Present	VB/VBZ, VBZ/VB	1000	see:sees return:___
Verbs	Past/3rd Person Singular Present	VBD/VBZ, VBZ/VBD	1000	saw:sees returned:___

Table 1: Test set patterns. For a given pattern and word-pair, both orderings occur in the test set. For example, if "see:saw return:___" occurs, so will "saw:see returned:___".

Ranking of Semantic Lexical Analogies

clothes:shirt :: dish:bowl

- 79 fine-grained word relations, where 10 are used for training and 69 testing
- Each relation is exemplified by 3 or 4 gold word pairs.
- Given a group of word pairs that supposedly have the same relation, the task is to order these pairs according to the degree to which

Method

Resolution of Syntactic Lexical Analogies

A:B::C:??

- Compute $\overrightarrow{D} = \overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C}$
- Retrieve the word whose embedding vector has the greatest cosine similarity to $\overrightarrow{D}$

Ranking of Semantic Lexical Analogies

score(A:B::C:D) ?

- Rank candidates by relational similarity

$$cos(\overrightarrow{D}, \overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C})$$

Results

Resolution of Syntactic Lexical Analogies

A:B::C:??

Mikolov's word embeddings capture significantly more syntactic regularity than the LSA vectors, and achieves around 40% accuracy

Ranking of Semantic Lexical Analogies

score(A:B::C:D) ?

Outperforms previous work

Summary

Analogical reasoning to evaluate quality of word embeddings

Ranking retrieval candidates

$$cos(\overrightarrow{D}, \overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C})$$

Resolution by retrieval

$$\overrightarrow{D} = \overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C}$$

Psychometric Analogy Tests

SAT tests

- Word analogy tests commonly used in assessments of linguistic and cognitive ability, included in US college admission test.
- Requires identifying ***fine-grained semantic differences*** between word pairs that belong to the same coarse-grained relation.

ushio et al. 2021

Query:		word:language
Candidates:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	**note:music**
	(4)	tale:story
	(5)	week:year

"a year consists of weeks" like "language consists of words", but the week-year pair is less similar to word-language than note-music.

New Lexical Semantic Analogy Benchmark

- The Google analogy dataset is the benchmark used by Mikolov

- BATS includes a larger number of concepts and relations, which are split into four categories: lexicographic, encyclopedic, and derivational and inflectional morphology

- SAT: 374 word analogy problems, consisting primarily of problems from US college admission SAT tests.

Dataset	Data size (val / test)	No. candidates	No. groups
SAT	37 / 337	5	2
UNIT 2	24 / 228	5,4,3	9
UNIT 4	48 / 432	5,4,3	5
Google	50 / 500	4	2
BATS	199 / 1799	4	3

- UNIT 2 - similar to SAT benchmark, but aimed at children in grades 4 to 12 from the US school system (i.e. from age 9 onwards).

- UNIT 4 - 5 difficulty levels, low-advanced level = SAT tests, high-

Retrieval-Based Analogy Completion

Given a query word pair (h_q, t_q) and a list of candidate answer pairs (h_i, t_i), the goal is to find the candidate answer pair that has the most similar relation to the query pair.

Analogy Solving

Used pretrained Language Models (LMs) to solve analogy problems without fine-tuning (Zero Shot Setting)

Query:		word:language
Candidates:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	**note:music**
	(4)	tale:story
	(5)	week:year

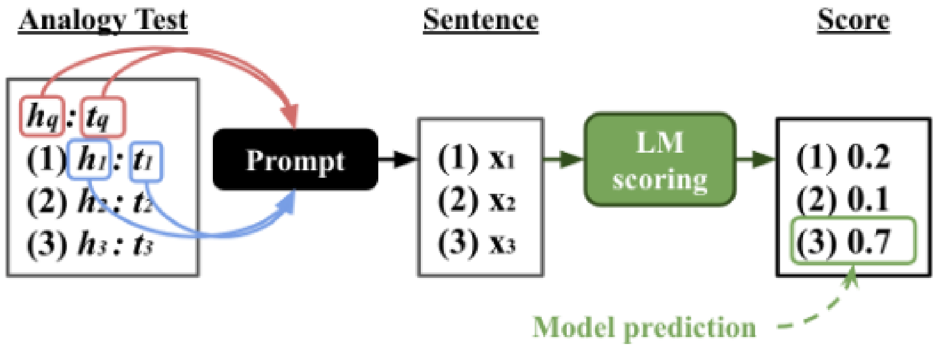
Model

Model: Pretrained LM + Prompt

Zero-shot setting

Each quadruplet is converted into a sentence that is input to the LM.

T(“word”, “language”, “note”, “music”)
$\Rightarrow$
“word is to language as note is to music”



The resulting sentences are then ranked using a scoring function: Perplexity, Pointwise-Mutual Information (PMI) or Marginal Likelihood Biased Perplexity (mPPL)

Perplexity

x, the input sentence

$Pauto(x \mid xj-1)$, the likelihood from an autoregressive LM's next token prediction.

Perplexity (sentence fluency)

$$f(x) = exp(-\sum_{j=1}^{m} log\ Pauto(x \mid xj-1))$$

PMI

n, the number of candidates

PMI inspired scoring (difference between conditional likelihood and the marginal likelihood)

$$sPMI(D, C|A, B) = logP\ (D|C, A, B) - \alpha \cdot log\ P(D \mid A, B)$$

mPPL

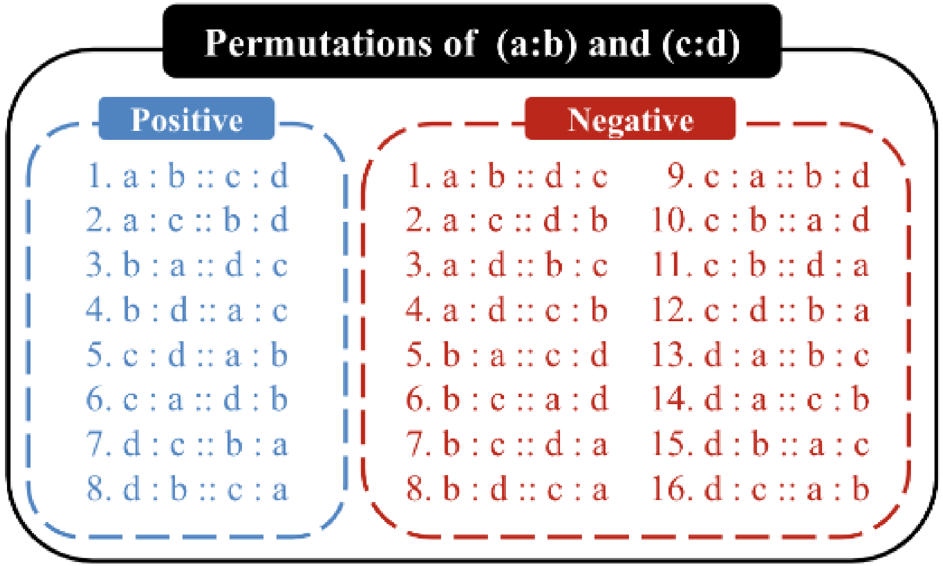
Extends perplexity with two bias terms

$$S_{mppl}(D, C|A, B) = log\ S_{PPL}(D, C|A, B) - \alpha_t \cdot log\ P(D \mid A, B) - \alpha_t \cdot log\ P(C \mid A, B)$$

Scoring

- Given some input, compute the score for its 8 positive and 16 negative permutation and aggregate these scores

- Select the candidate with highest score



Models and Baselines

- Three Encoders: BERT, RoBERTa, GPT-2
- Word embedding Models: Word2Vec, GloVe, FastText
  - Represent pairs by the difference between their embeddings (A-B, C-D)
  - Select candidate with highest cosine similarity to the query
- Random Baseline
- Select candidate for which word pair PMI score is highest (ignore query)

Results

Accuracy on each dataset

- RoBERTa and GPT-2 consistently outperform BERT
- smPPL achieves substantially better results than sPMI or sPPL in most cases
- Scores are lower for SAT problems (harder benchmark)

	Model	Score	Tuned	SAT	U2	U4	Google	BATS	Avg
LM	BERT	s_{PPL}		32.9	32.9	34.0	80.8	61.5	48.4
			✓	39.8	41.7	41.0	86.8	67.9	55.4
		s_{PMI}		27.0	32.0	31.2	74.0	59.1	44.7
			✓	40.4	42.5	27.8	87.0	68.1	53.2
		s_{mPPL}	✓	41.8	44.7	41.2	88.8	67.9	56.9
	GPT-2	s_{PPL}		35.9	41.2	44.9	80.4	63.5	53.2
			✓	50.4	48.7	51.2	93.2	75.9	63.9
		s_{PMI}		34.4	44.7	43.3	62.8	62.8	49.6
			✓	51.0	37.7	50.5	91.0	79.8	62.0
		s_{mPPL}	✓	**56.7**	50.9	49.5	95.2	**81.2**	66.7
	RoBERTa	s_{PPL}		42.4	49.1	49.1	90.8	69.7	60.2
			✓	53.7	57.0	55.8	93.6	80.5	68.1
		s_{PMI}		35.9	42.5	44.0	60.8	60.8	48.8
			✓	51.3	49.1	38.7	92.4	77.2	61.7
		s_{mPPL}	✓	53.4	**58.3**	**57.4**	93.6	78.4	**68.2**
WE	FastText	-		47.8	43.0	40.7	**96.6**	72.0	60.0
	GloVe	-		47.8	46.5	39.8	96.0	68.7	59.8
	Word2vec	-		41.8	40.4	39.6	93.2	63.8	55.8
Base	PMI	-		23.3	32.9	39.1	57.4	42.7	39.1
	Random	-		20.0	23.6	24.2	25.0	25.0	23.6

Summary

Like Mikolov uses analogies to test existing neural representations

- Harder benchmark (SAT)
- Representations from pretrained encoders and language models (Bert, Roberta, GPT2 representations)
- Rank candidates using scoring functions inspired from analogical reasoning

$$sPMI(D, C|A, B) = logP\ (D|C, A, B) - \alpha \cdot log\ P(D \mid A, B)$$

$$S_{mppl}(D, C|A, B) = log\ S_{PPL}(D, C|A, B) - \alpha_t \cdot log\ P(D \mid A, B) - \alpha_t \cdot log\ P(C \mid A, B)$$

- ***Zero Shot:*** no training data needed (but restricted to Analogy ranking)

Learning Representations of Analogy

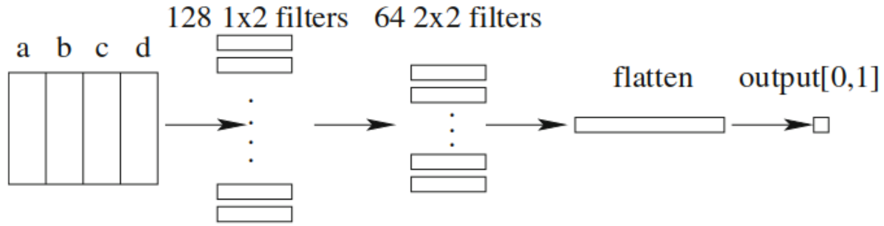
Learn word representations which capture analogy

Lim et al. 2019

Learning Representations of Analogy

Classifier

- CNN Encoder
- Pre-trained GloVe embeddings for words
- Stack the Glove vectors for A, B, C and D into a matrix (aka image)
- Training: 10 fold cross-validation, maximise cross entropy



Learning Representations of Analogy

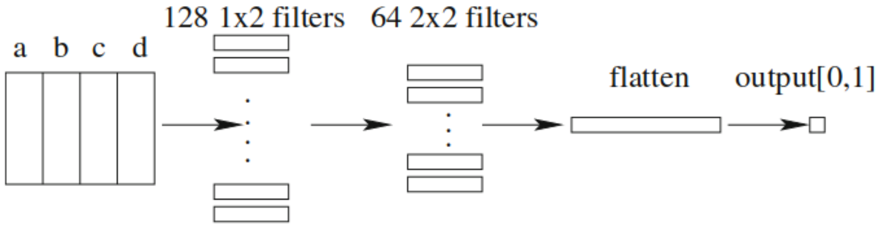
Classifier

Layer 1

- Filters slide over (a,b) and (c,d) separately
- Output two activation maps (M$_{AB}$, M$_{CD}$) which represent their respective differences/similarities

Layer 2

- Filters slide over M$_{AB}$, M$_{CD}$ jointly, comparing the two representations



Last layer

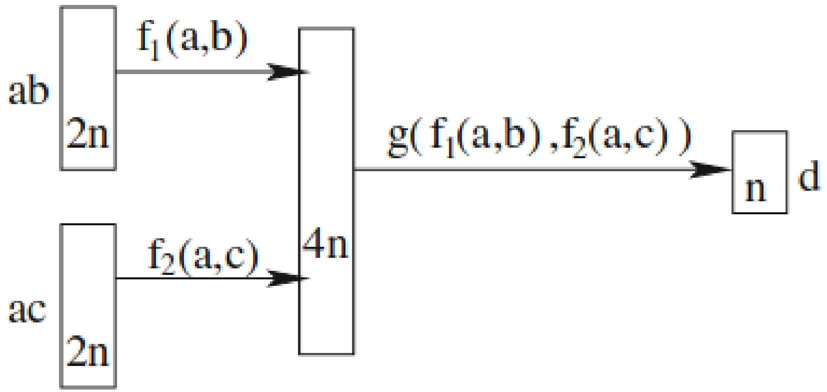
- Outputs a value between 0 an 1

Learning Representations of Analogies

Retrieval-Based Analogy Resolution

Hierarchical encoder

- Encode *a;b* and *a;c*
- Project the concatenated result onto a vector *d*
- Retrieve word whose vector is closest to *d*



Data

Dataset

- Google dataset: 19,544 analogies
- Diverse relations such as
  - capital-countries
  - country-currency
  - opposite

Data Augmentation

- Use 8 permutation properties to create 19,544 × 8 = 156,352 valid analogies
- Permute the first 2 elements to create 469,056 examples invalid analogies
- Total data: 625,408 examples (156K valid analogies, 469K invalid analogies)

Classification Results (Accuracy)

Table 2. Accuracy of the CNN for analogy classification: impact of word embedding dimensions and the number of epochs.

#epochs	Average accuracy (std dev.)			
	50	100	200	300
1	83.9% (± 6.4)	81.1% (± 8.5)	75.5 (± 1.2)	80.26 (± 11.13)
3	81.19% (± 6.87)	81.40% (±12.01)	84.27% (± 8.19)	83.84% (± 7.77%)
5	90.68% (± 5.77)	93.07% (± 6.83)	93.19% (± 6.21)	95.22% (± 4.90%)
10	91.02% (± 7.67)	96.79% (± 5.05)	99.24% (± 1.17)	99.34% (± 0.79%)

- Almost perfect accuracy

Retrieval Results (Accuracy)

- The best overall performance is given by 100 dimensions at 79.0% of accuracy
- Outperforms previous work

Table 3. Comparing the effectiveness of the neural network and the formula 3*CosMul* for all categories and each category. The total number of analogies is 19,544, and the "Common Capital" category has 506 analogies.

	Neural network regression				3CosMul			
	50	100	200	300	50	100	200	300
Overall	63.9%	79.0%	75.4%	71.4%	36.2%	56.7%	65.0%	68.1%
Common capitals (506)	96.3%	98.8%	97.4%	96.9%	63.8%	80.0%	86.9%	88.9%
All capitals (4524)	86.7%	97.1%	97.1%	90.9%	50.0%	77.0%	87.5%	90.1%
Currencies (866)	50.2%	63.4%	61.2%	56.3%	5.0%	15.0%	22.5%	24.4%
US cities (2467)	32.4%	53.5%	62.0%	59.5%	6.9%	17.0%	29.8%	36.5%
Gender (506)	53.2%	44.3%	40.7%	34.4%	61.5%	79.4%	86.6%	88.6%
Adj to adverb (992)	34.5%	55.1%	31.9%	30.1%	10.8%	22.0%	22.9%	23.4%
Opposite (812)	24.9%	43.1%	26.3%	23.3%	6.2%	17.1%	21.5%	25.4%
Comparative (1332)	74.6%	89.2%	85.6%	83.2%	41.4%	71.9%	79.8%	83.3%
Superlative (1122)	73.1%	86.4%	78.9%	76.2%	18.6%	50.1%	67.5%	73.7%
Base to gerund (1056)	43.9%	78.3%	67.9%	70.3%	35.2%	65.6%	68.1%	71.0%
Nationalities (1599)	93.5%	94.4%	96.3%	94.3%	84.7%	89.1%	94.1%	94.6%
Gerund to past (1560)	52.2%	80.8%	69.5%	66.7%	27.3%	53.1%	59.6%	62.5%
Plurals (1332)	78.3%	87.1%	88.1%	74.6%	48.2%	68.5%	74.1%	76.5%
Base to 3^{rd} person (870)	46.2%	74.8%	59.2%	56.4%	28.8%	59.1%	64.9%	68.4%
MSE (train)	0.1	0.07	0.06	0.05				
MSE (test)	0.1	0.07	0.06	0.05				

Classifying Morphological Analogies

✓ *cats:cat :: trees:tree*

✓ *chats:chat :: arbres:arbre*

~~*chats:chat :: chanter:chante*~~

Language	Train	Dev	Test
Arabic	373240	7671	555312
Finnish	1342639	22837	4691453
Georgian	3553763	67457	8368323
German	994740	17222	1480256
Hungarian	3280891	70565	66195
Maltese	104883	3775	3707
Navajo	502637	33976	4843
Russian	1965533	32214	6421514
Spanish	1425838	25590	4794504
Turkish	606873	11518	11360

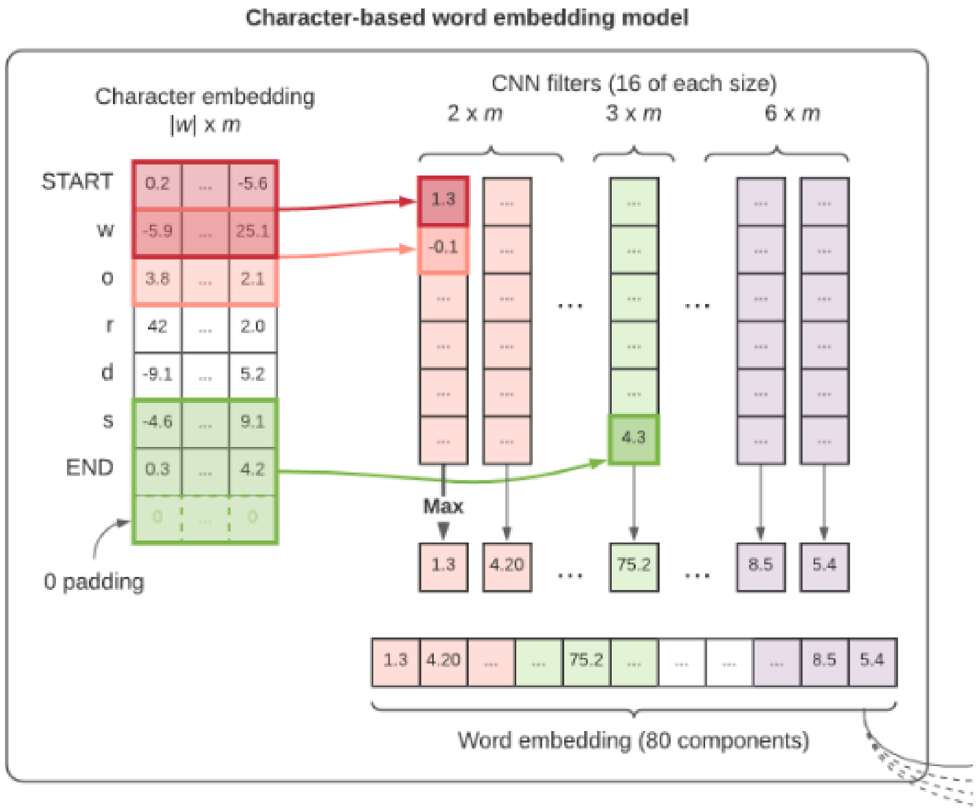
Alsaidi et al. 2021

Representing words using Character Embeddings

- A word is represented by a concatenation of character embeddings
- Character embeddings are vectors trained jointly with the classifier using a CNN architecture

  80 Filters of size 2 to 6 capture subwords (aka morphemes) of size 2 to 6 characters

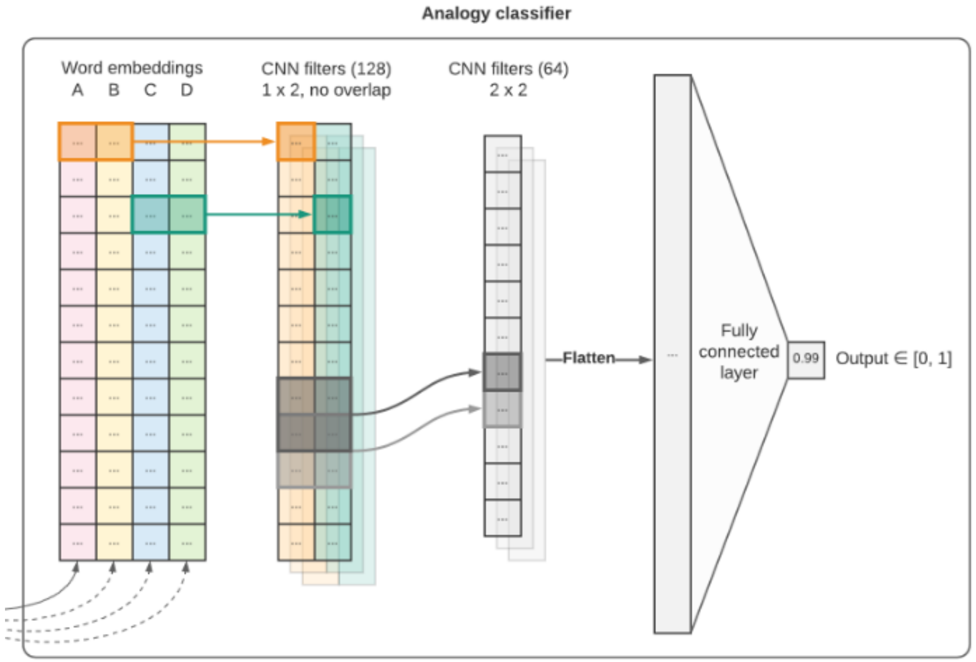
  Max pooling projects each character embedding to a vector of size 80



Classifier

Convolutional Neural Network (CNN)

- Compares Ai/Bi and Ci/Di
- Compares $\Delta(A, B), \Delta(C, D)$
- Classify



Data

Sigmorphon2016 Data

- Ten languages with rich morphology
- Arabic (Romanized), Finnish, Georgian, German, Hungarian, Maltese, Navajo, Russian, Spanish, and Turkish
- Triples of the form (lemma, features, form)
  E.g., ***(do, present participle, doing)***

Creating Analogies

- (lemma, form) pairs which share the same features

(do, present participle, doing)
(play, present participle, playing)
→ ***do:doing :: play:playing***

- **+ *Data Augmentation***

A : B :: A' : B' →

A′ : B′ :: A : B , A : B :: A : B ...

Training and Results

- Encoder (Word embeddings) and classifier are learned jointly using binary cross-entropy
- Outperforms previous analogy based approaches

Language	CNN	Best baseline according to [8]
Arabic	**98.75**	93.33 (Lepage)
Finnish	93.57	**93.69** (Kolmo)
Georgian	**99.56**	99.35 (Kolmo)
German	**99.56**	98.84 (Kolmo)
Hungarian	**99.32**	95.71 (Kolmo)
Maltese	**97.93**	96.38 (Kolmo)
Navajo	**99.82**	86.87 (Lepage)
Russian	**99.61**	97.26 (Lepage)
Spanish	**97.37**	96.73 (Kolmo)
Turkish	**99.77**	89.45 (Kolmo)

Solving Morphological Analogies

A:B ::C → ??

dance:dancer :: run → runner

Data

- (Sigmorphon 2019): Arabic, English, French, German, Hungarian, Portuguese, Russian, and Spanish
- Data Augmentation

A:B :: C:D → A : C :: B : D, D : B :: C : A, C : A :: D : B ...

Chan et al. 2022

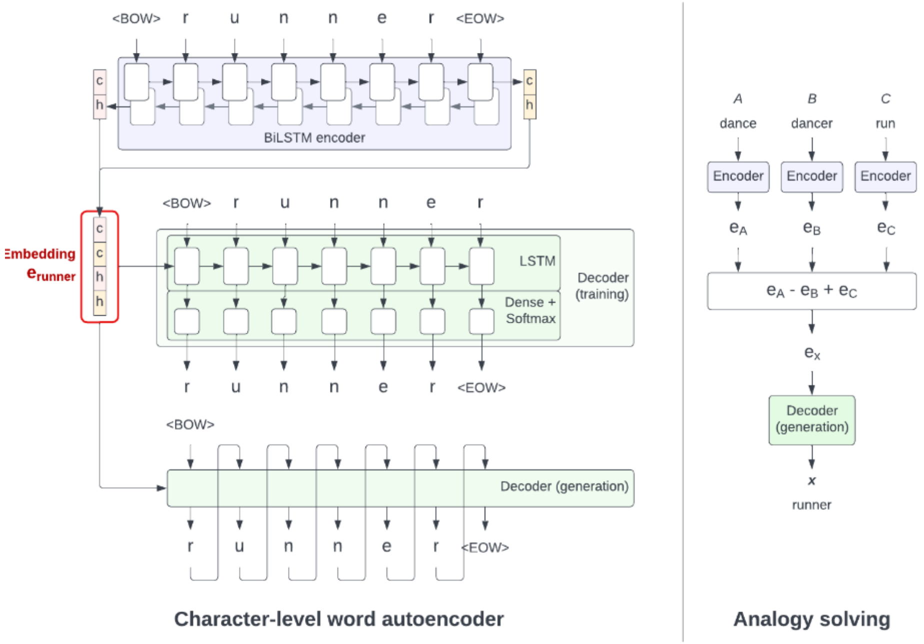
Model

Word Representations

- Learned using a character level Bi-LSTM auto-encoder

Encoder-Decoder Model

- Encode A, B, C into embeddings $\overrightarrow{A}, \overrightarrow{B}, \overrightarrow{C}$
- $\overrightarrow{D} = \overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C}$
- Decode output word from $\overrightarrow{D}$



Results

- Evaluation Metrics
  - L_p - Normalized Levensthein distance
  - Acc. - ratio of correct outputs
- Retrieval model performs better
- Lower performance on irregular morphology
- Morphological features might help improve results

Language	Score	Ours	Alea	Kolmo	ANNr
Arabic	L_p	**54.51**	23.72	45.31	–
	Acc.	**12.50**	2.56	3.81	71.80 ± 2.51
English	L_p	**91.58**	88.34	86.75	–
	Acc.	**59.80**	59.65	46.93	94.40 ± 0.67
French	L_p	86.43	80.07	**89.32**	–
	Acc.	51.30	**57.64**	54.49	91.84 ± 0.83
German	L_p	**89.39**	82.76	87.47	–
	Acc.	**52.80**	50.84	48.97	76.95 ± 1.15
Hungarian	L_p	**80.32**	60.72	75.47	–
	Acc.	25.50	**27.80**	23.48	80.42 ± 1.30
Portuguese	L_p	**94.38**	87.97	93.47	–
	Acc.	74.00	**80.06**	71.28	89.30 ± 2.38
Russian	L_p	82.29	63.52	**82.78**	–
	Acc.	33.80	**37.15**	33.44	72.65 ± 1.96
Spanish	L_p	**89.39**	79.49	88.56	–
	Acc.	60.09	**65.02**	58.59	93.01 ± 2.38

Summary

Dedicated encoders

Learn word embeddings which capture analogy

Architectures

- CNN for classification
- hierarchical dual encoder for resolution by retrieval
- Encoder-decoder for resolution by generation

Sentential Analogies

(question,answer)

Question-Answer Analogical Quadruplets

- QA by retrieval]

- Assumes an ***analogical relation between Q/A pairs of the same type (who, when, where)***

- Learning representations for QA pairs which capture this relation helps improve retrieval-based QA

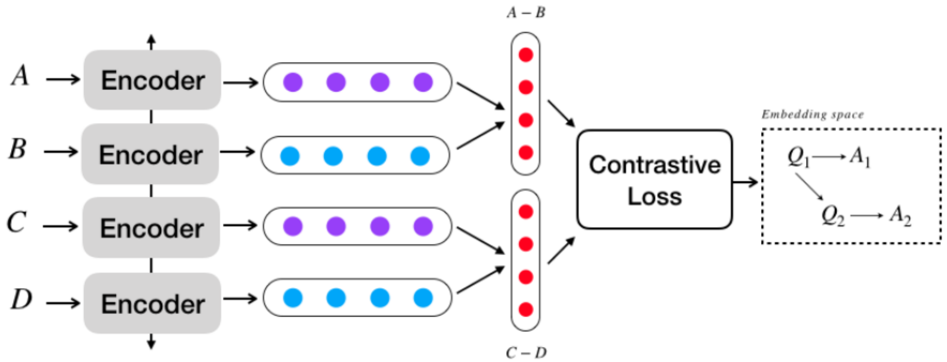
	"Where" questions
Sentence A	"Where was Abraham Lincoln born?"
Sentence B	"On February 12, 1809, Abraham Lincoln was born Hardin County, Kentucky
Sentence C	"Where was Franz Kafka born?"
Sentence D	"Franz Kafka was born on July 3, 1883 in Prague, Bohemia, now the Czech R
	"Who" questions
Sentence A	"Who made the rotary engine automobile?"
Sentence B	"Mazda continued work on developing the Wankel rotary engine."
Sentence C	"Who discovered prions?"
Sentence D	"Prusiner won Nobel prize last year for discovering prions"
	"When" questions
Sentence A	"When was Leonardo da Vinci born?"
Sentence B	"Leonardo da Vinci was actually born on 15 April 1452 [...] "
Sentence C	"When did Mt St Helen last have significant eruption?"
Sentence D	"Pinatubo's last eruption [...] as Mt St Helen's did when it erupted in 1980."

A. Diallo et al., 2019

Model

]

- Siamese network with 4 inputs
- bi-GRU with max pooling on each input
- Trained to minimize the difference between two analogous pairs in the embedding space



$$\mathcal{L}_W = y\mathcal{L}_+(AB, CD)$$

$$+(1-y)\mathcal{L}_-(AB, CD)$$

$$sim(AB, CD) = cos(A - B, C - D)$$

$$\mathcal{L}_+(A - B, C - D) = (1 - sim(AB, CD))^2$$

$$\mathcal{L}_-(A - B, C - D) = max(sim(AB, CD) - m)^2$$

Creating Analogical Quadruples

- Three categories of questions: ”Who”, ”When” and ”Where”
- Positive example: Q/A pair and prototype of the same category
- Negative example: Q/X pair and prototype of the same category, X is not the answer to Q

	WikiQA			TrecQA		
type	train	dev	test	train	dev	test
”Who”	119	15	34	190	11	8
”When”	86	11	16	116	13	19
”Where”	71	17	22	96	9	11
Comb.	276	43	72	402	33	38

Results

Compare ranking by cosine similarity of the difference vectors for other vectors with

- Averaging word vectors
- Sentence vectors

Explicitly constraining structural analogies in the Q/A embedding space leads to better results over distance-only embeddings

	Model	WikiQA	
		MAP	MRR
W.E	Glove	0.464	0.475
	Word2Vec	0.4329	0.453
S.E	InferSent	0.399	0.404
	Sent2Vec	0.481	0.486
	This work	**0.6771**	**0.6841**

Similarity score used for retrieval:

$$sim(AB, CD) = cos(A - B, C - D)$$

Summary

- Q/A embeddings trained to capture analogical structure
- These embeddings are shown to improve retrieval-based QA based on standard word and sentence embeddings
- Limited scope
  - where, who, when
  - small dataset

Sentential Analogies

(syntax and semantics)

Analogy Solving by Retrieval

Goal

Do sentence representation spaces created by neural approaches capture sentence level analogies ?

Test: Analogy Solving by Retrieval

Retrieve D whose embedding maximises

$$cos(\overrightarrow{D}, \overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C})$$

Zhu et al. 2020

Sentence Pairs

	S_A	S_B
Common Capital Cities	They traveled to **Havana**.	They took a trip to **Cuba**.
All Capital Cities	I've never been to **Amman**.	I've never been to **Jordan**.
Currencies	The economy in **Japan** was great.	The **yen** appreciated due to the strong economy.
City in State	They go down to **Chandler**.	They go down to **Arizona**.
Man – Woman	The **man** makes wooden crafts and arts.	The **woman** makes wooden crafts and arts.
Comparative	The second article was **long**.	The second article was **longer** than the first one.
Nationality Adjective	The man from **Egypt** tapped his cheek.	The **Egyptian** man tapped his cheek.
Opposites	It's **possible** to measure it.	It's **impossible** to measure it.
Plurals	The Harvard data examined one **city**.	The Harvard data examined six **cities**.
Verb Conjunction	Duke will **play** better this year.	Duke **plays** better this year.

Category	Sentence Pairs	Analogies
Common Capital City	138	9,453
All Capital Cities	928	430,128
City in State	402	80,601
Currency	150	11,175
Gender	126	7,875
Comparative	466	108,345
Opposite	513	131,328
Nationality Adjective	205	20,910
Plural	512	130,816
Verb Conjugation	451	101,475
Entailment	673	226,128
Negation	511	130,305
Passivization	256	32,640
Objective Clause	563	158,203

- Sentence pairs where one word is replaced with a word from the Google word analogy dataset
- Sentence pairs that share common semantic (entailment, negation, passivization etc.) or syntactic (comparisons, opposites, plurals ...) relations

Zhu et al. 2020

Retrieval candidates

Table 2: Example of candidate sets for relation-based analogy

	Entailment	Negation
S_A	The man is heaving barbells.	There is no deer jumping a fence.
S_B	The man is lifting barbells.	A deer is jumping over the fence.
S_C	A man is singing a song and playing the guitar.	There is no boy hitting the football.
Positive Candidate	A man is singing and playing the guitar.	A boy is hitting the football.
Not Negation	A man is not singing and not playing the guitar.	A boy is not hitting the football.
Random Deletion	A man is the guitar.	is the football.
Random Masking	A [MASK] is singing and playing the guitar.	A [MASK] is [MASK] the football.
Span Deletion	A man is singing the guitar.	A boy the football.
Word Reordering	and playing the guitar A man is singing.	The football a boy is hitting.

One correct candidate

Several *challenging distractors* created by adding or removing negation, random word delection, random masking (replace a word with meaninglss token), span deletion and word reordering

Models and Results

Models

- Average of Glove embeddings
- Concatenation of Discrete Cosine Transform coefficients embeedings
- Skip-Thought Vectors
- Quick-Thought Vectors
- GenSen
- InferSentV1, InferSentV2
- USE-DAN, USE-Transformer
- CLS, avg on BERT, XLNet, RoBERTa SBERT

Results

Lexical transforms

- Removing A, B and C from the retrieval set matters

Relational Analogies

- InferSentV2 and GenSen models achieve the highest results
- The large version of XLNet achieves the lowest accuracy.

Summary

- Similar to Mikolov, extended to sentences
- Analogy resolution by retrieval
- Extensive comparison of existing word and sentence encoders

Sentential Analogies

(syntax and semantics)

Analogy Solving by Generation

Solve sentence analogies by generating the solution rather than retrieving the best candidate from a pool of retrieval candidates.

Encoder

- Learns an analogical representation $\overrightarrow{D}$ of D from A, B and C

Decoder

- Trained to generate D from $\overrightarrow{D}$

Wang et al. 2020

Predicting the solution vector to an analogical equation

Experiment with three ways of combining the input vectors $\overrightarrow{A}, \overrightarrow{B}, \overrightarrow{C}$

- Concatenation
  $\overrightarrow{A} \cdot \overrightarrow{B} \cdot \overrightarrow{C}$
- Summation
  $\overrightarrow{A} + \overrightarrow{B} + \overrightarrow{C}$
- Arithmetic analogy
  $\overrightarrow{B} - \overrightarrow{A} + \overrightarrow{C}$

ABC vector learned using MSE Loss

$$L_{mse} = \frac{1}{K} \sum_{k=1}^{K} (\overrightarrow{ABC}_k, \overrightarrow{D}_k)^2$$

Pretrained Decoder

- Input: a pretrained sentence vector (SBERT, fastText), $\overrightarrow{s}$
- RNN trained to reconstruct input sentence using two losses
  - Cross entropy loss - similarity between predicted and expected token
  - Regression loss - similarity between the hidden state of the recurrent units and the embeddings of the expected token at every time step

Classification Loss

$$LCE = \frac{1}{N}\sum_{n=1}^{N} log\ p(w_n \mid c)$$

Regression Loss

$$LMSE = \frac{1}{K}\sum_{k=1}^{K}(v_k - v(D)_k)^2$$

with K the dimension of the word embeddings.

Data

Training Data for Decoder

- English sentences from Tatoeba corpus
- 79,171 sentences with an average length of 7 words

Data	Number of		
	sentences	words/sent.	characters/sent.
Training	63,336	6.7 ± 1.6	28.5 ± 8.0
Validation	7,917	6.7 ± 1.6	28.4 ± 8.0
Testing	7,918	6.7 ± 1.6	28.5 ± 8.0
Total	79,171		

Analogical Data

- 5,607 labeled analogical equations between sentences, which include formal and semantic analogies between chunks.

Data	Number of			
	analogies	sentences	words/sent.	characters/sent.
Training	3,364	3,185	7.1 ± 1.2	27.0 ± 5.7
Validation	1,121	1,769	7.1 ± 1.1	26.6 ± 5.6
Testing	1,121	1,667	7.0 ± 1.1	26.3 ± 5.6
Total	5,607			

Results

Resolution	Composition	Edit Distance in words	Edit Distance in char.	Jaccard Similarity in words	Jaccard Similarity in characters	Accuracy (%)	BLEU	METEOR
Vector offset method		1.3 ± 1.2	5.0 ± 5.2	0.84 ± 0.14	0.85 ± 0.15	41.90	0.75 ± 0.01	0.50
Linear regression	concatenation	0.7 ± 1.4	2.5 ± 5.1	0.92 ± 0.15	0.93 ± 0.13	74.24	0.87 ± 0.02	0.59
	summation	1.6 ± 2.2	5.2 ± 6.9	0.82 ± 0.22	0.85 ± 0.18	52.41	0.72 ± 0.02	0.49
	arithmetic nlg.	**0.4 ± 1.1**	**1.6 ± 4.3**	**0.95 ± 0.12**	**0.96 ± 0.10**	**83.24**	**0.91 ± 0.01**	**0.64**

- Analogical embeddings (B - A + C) outperform the vector offset method based on standard sentence embeddings.

Fine-tuning Pretrained Models on Analogies

- BART Sequence-to-Sequence (S2S) Model
- GPT-2 Language Model

Fine-tuning BART S2S Model on masked data

Any span

*he will **[mask]** will come. :: i have no time tomorrow. : i have no time.*

Any term

*he will come tomorrow. : he will come. :: **[mask]** : i have no time.*

Term D

*he will come tomorrow. : he will come. :: i have no time tomorrow. : **[mask]***

MLM Objective

Wang et al. 2022

Fine tune GPT-2 Language Model to generate term D

he will come tomorrow. : he will come. :: i have no time tomorrow. : ***[mask]***

[mask] $\rightarrow$ *i have no time*

Data

Phrase Analogy (PA)

to say : want to say :: to go out : want to go out

- Based on 3,003 sentences with an average length of 25 words
- 25,310 phrases of length between 2 and 6
- Analogy: 2 pairs of phrases which illustrates the same syntactic transformation
- ***Training Data:*** 1.5M phrase analogies with an average length of 3 words
- ***Test Data:*** 1K instance
- ***OOD Test data:*** Sentence analogies from the Tatoeba corpus, with sentence length from from 2 to 10. 1K instances.

Results: S2S vs LM

Data	Model	Masking scheme	Acc (%)	Levenshtein distance in chars
PA	GPT-2	-	**99.7**	**0.01±0.01**
	BART	Any span	97.5	0.05±0.03
		Any term	97.0	0.05±0.03
		Term *D*	50.9	0.58±0.07
SA	GPT-2	-	4.2	12.92±0.83
	BART	Any span	11.4	4.28±0.38
		Any term	**44.4**	**2.85±0.29**
		Term *D*	12.0	3.17±0.30

Performance decreases on OOD

In-Domain Data (PA)

- LM fine-tuning performs best
- D masking for ED performs worst (lack of right context ?)

OOD Data (SA)

- S2S performs better than LM on OOD data

Summary

- Two methods for sentence analogy resolution by generation
  - Encoder-decoder with custom encoder and decoder
  - BART fine tuning using masked language modelling objective
- limited to short sentences
- Generalises poorly to OOD

Conclusion

Word2vec, BERT, GPT-2, BART

- creates representations that have been shown to perfom well on a wide variety of tasks
- Analogies used to analyse the quality of neural representations

Custom encoders developed to support analogy classification and resolution

- How well do these encoders perform on other tasks ?
- Do these encoders allow for better generalisation ?

Bibliography

S. Afantenos, S. Lim, H. Prade and G. Richard, Theoretical study and empirical investigation of sentence analogies. Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria.

Allen, T. Hospedales, Analogies explained: Towards understanding word embeddings, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 223–233

S. Alsaidi, A. Decker, P. Lay, E. Marquer, P-A Murena, M. Couceiro. (2021a). A Neural Approach for Detecting Morphological Analogies. In: The 8th IEEE International Conference on Data Science and Advanced Analytics (DSAA). Porto/Online, Portugal. url:

P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, in: Transactions of the Association for Computational Linguistics, 2017, p. 135–146.

Z. Bouraoui, S. Jameel, S. Schockaert, Relation induction in word embeddings revisited, in: COLING, 1627-1637, Assoc. Computat. Ling., 2018.

S. R. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2015.

K. Chan, S. P. Kaszefski Yaschuk1, C. Saran1, E. Marquer and M. Couceiro. Solving Morphological Analogies Through Generation

Chollet 2019, On the measure of intelligence, 2019.

A. Diallo, M. Zopf, J. Fürnkranz, Learning analogy-preserving sentence

W. B. Dolan, C. Brockett, Automatically constructing a corpus of sentential paraphrases, in: Proceedings of the Third International Workshop on Paraphrasing (IWP2005), 2005.

A. Drozd, A. Gladkova, S. Matsuoka, Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen, in: COLING, 2016, pp. 3519–3530.

K. Guu, Tatsunori B. Hashimoto, Yonatan Oren, Percy Liang. TACL 2018. Generating Sentences by Editing Prototypes

E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, in: Proc. of LREC, 2018.

Hofstadter 2001, Analogy as the Core of Cognition, MIT Press, 2001, pp. 499–538.

Y. Lepage, De l’analogie rendant compte de la commutation en linguistique, Habilit. à Diriger des Recher., Univ. J. Fourier, Grenoble (2003).

S. Lim, H. Prade and G. Richard, "Solving Word Analogies: A Machine Lerning Perspective" G. Kern-Isberner and Z. Ognjanovic (Eds.): ECSQARU 2019, LNAI 11726, pp. 238–250, 2019.

Y. Liu and Y. Lepage. 2021. Covering a sentence in form and meaning with fewer retrieved sentences. In Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation, pages 513–522, Shanghai, China. Association for Computational Lingustics.

T. Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. Proceedings of International Conference on Learning Representations (ICLR).

T. Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26 (NIPS 2013). pages 3111–3119.

M. Mitchell Abstraction and analogy-making in artificial intelligence,, Annals of the New York Academy of Sciences, 2021.

J. Peyre et al. “Detecting Unseen Visual Relations Using Analogies”. In: ICCV 2019. IEEE, 2019, pp. 1981–1990

H. Prade, G. Richard, From analogical proportion to logical proportions, Logica Univers. 7 (2013) 441–505.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, B. Webber, The Penn Discourse TreeBank 2.0., in: LREC 08, 2008.

F. Sadeghi et al. “Visalogy: Answering Visual Analogy Questions”. In: NIPS 2015. 2015, pp. 1882–1890.

V. Taillandier, L. Wang and Y. Lepage. Réseaux de neurones pour la résolution d’analogies entre phrases en traduction automatique par l’exemple.

P. D. Turney, A uniform approach to analogies, synonyms, antonyms, and associations, in: COLING, 2008, pp. 905–912.

P. D. Turney, Distributional semantics beyond words: Supervised learning of analogy and paraphrase, TACL 1 (2013) 353–366

A. Ushio, L. Espinosa Anke, S. Schockaert, J. Camacho-Collados, BERT is to NLP what AlexNet is to CV: Can pre-trained language models identify analogies?, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 3609–3624. doi:10.18653/ v1/2021.acl-long.280.

L. Wang, Y. Lepage. Masked prompt learning for formal analogies beyond words. IARML Workshop , IJCAI-ECAI, July 23-29, 2022, Wien.

L. Wang, Y. Lepage, Vector-to-sequence models for sentence analogies, in: 2020