UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Semantic Specialization of Distributional Representation Models

November 4, 2019
EMNLP 2019, Tutorial

Goran Glavaš
Uni Mannheim

Edoardo Ponti
Uni Cambridge

Ivan Vulić
Uni Cambridge

Tutorial Overview

1. **Introduction and Motivation:** distributional representation models (Static vs Non-static), lexical relations, external repositories, complementarity of information
(20 minutes, Ivan)

2. **Specialisation for Semantic Similarity**: similarity vs relatedness vs other relationships, joint versus retrofitting models, explicit retrofitting versus post-specialisation, evaluating for semantic similarity
(45 minutes, Ivan)

3. **Specialisation for LE and Other Relations**: specialisation for lexical entailment, embedding hierarchies in vector spaces, explicit versus post-specialisation for LE; specialization for other relations, evaluation
(35 minutes, Goran)

4. **Cross-lingual Transfer of Specialisation**: different approaches to target language specialisation, supporting the construction of lexical resources in resource-poor language; challenges with resource-low settings
(25 minutes, Goran)

5. **Specialisation of Contextualised Representation Models**: LIBERT, K-BERT, ERNIEs, etc.
(45 minutes, Edoardo)

6. **Challenges, Open Problems, Conclusions**
(10 minutes, Edoardo)

Tutorial Goals

This tutorial aims to provide (some) answers to the following main questions:

1) How to combine **continuous representations from text** with **discrete external knowledge** (primarily from linguistic resources)?

2) How to fuse the information for static and for contextualised distributional representations?

3) What are the modeling differences when specializing for relations with different properties (e.g., **symmetric** versus **asymmetric** relations)?

4) How to deal with incomplete information in external resources?

5) How to transfer the external knowledge **across languages**? How to deal with resource-poor languages with scarce or non-existent resources?

6) What classes of downstream applications are supported by different specializations?

… + Challenges + Open problems

Specialization: Motivation from Application(s)

Distributional representation models coalesce several types of information

- **True semantic similarity** versus (broader) **semantic relatedness** for dialogue?

User: I'm looking for a cheaper restaurant
`inform(price=cheap)`
System: What kind of food?
User: English, in eastern Cambridge
`inform(price=cheap, food=British, area=east)`
System: The Green Man is the best choice
User: Where is it?
`inform(price=cheap, food=British, area=east);`
`request(address)`
System: The Green Man is at 59 High St, Grantchester

Specialization: Motivation from Application(s)

Distributional representation models coalesce several types of information

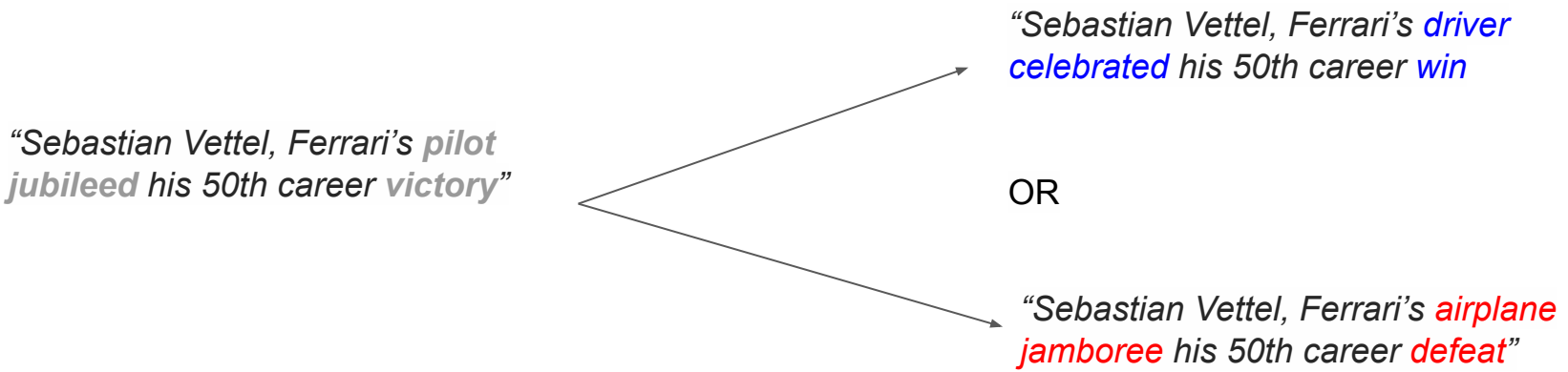
- **True semantic similarity** versus (broader) **semantic relatedness** for dialogue?

User: I'm looking for a cheaper restaurant
`inform(price=expensive)`
System: What kind of food?
User: English, in eastern Cambridge
`inform(price=expensive, food=Spanish, area=east)`
System: The Green Man is the best choice
User: Where is it?
`inform(price=expensive, food=Spanish, area=east); request(address)`
System: The Green Man is at 59 High St, Grantchester

Specialization: Motivation from Application(s)

Distributional representation models coalesce several types of information

- **True semantic similarity** versus (broader) **semantic relatedness** for text simplification?



Specialization: Motivation from Application(s)

- Nearest neighbors **before** and **after** specialization
  - impact on downstream tasks?

Word	**east**	**expensive**	**British**
Before	**west**	pricey	**American**
	north	**cheaper**	**Australian**
	south	costly	Britain
	southeast	overpriced	**European**
	northeast	**inexpensive**	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
	-	pricey	UK
	-	afford	Britain

Specialization: Motivation from Application(s)

Besides (more standard) NLP applications, specialization and specialized vectors found some unexpected applications in:

- Biomedical NLP
  [Crichton et al., BMC Bioinformatics-17,18; Chiu et al., BioNLP-19; Phan et al., ACL-19]
- Cognitive science and psychology
  [Richie et al., 2019]
- Debiasing word representations;
  [Lauscher et al., 2019]
- Detection of abusive language
  [Koufakou and Scott, WiNLP-19]
- Fact checking and verification
  [Jaradat et al., NAACL-18]
- Guiding NMT training
  [Wieting et al., ACL-19]

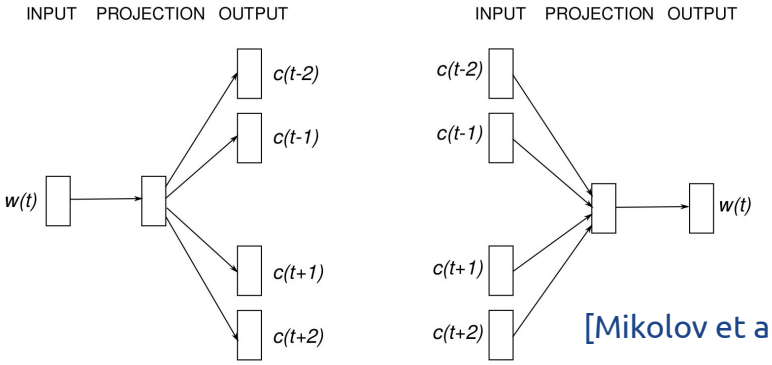
Specialization: Motivation from Lexical Semantics

Can we distinguish the type of relation between words from their distributional representations?

	car		driver		vehicle		wheel	
	GloVe	*fastText*	*GloVe*	*fastText*	*GloVe*	*fastText*	*GloVe*	*fastText*
car	1.0	1.0	0.81	0.69	0.86	0.69	0.67	0.47
driver	--	--	1.0	1.0	0.71	0.52	0.58	0.41
vehicle	--	--	--	--	1.0	1.0	0.60	0.44
wheel	--	--	--	--	--	--	1.0	1.0

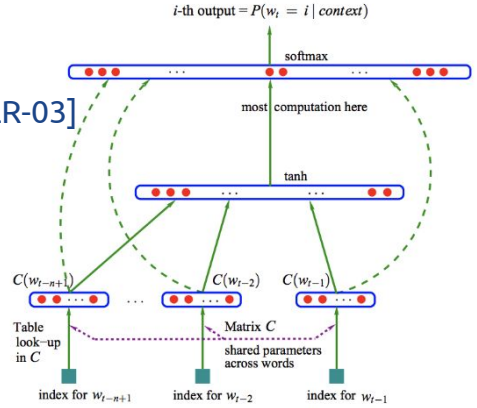
Learning from Distributional Signal

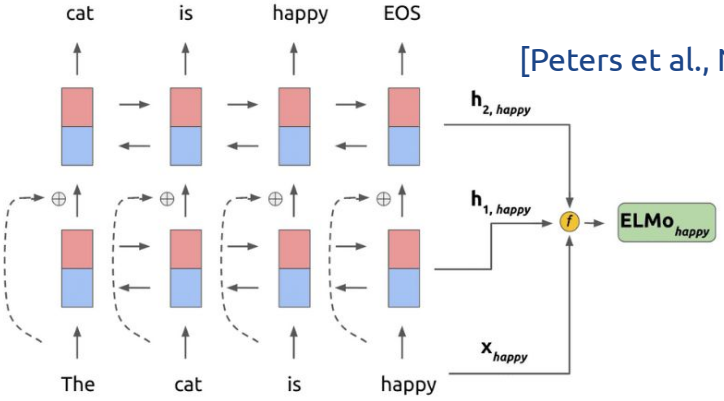
Different learning paradigms, but the underlying data and (“meaning-as-use”) assumptions are similar



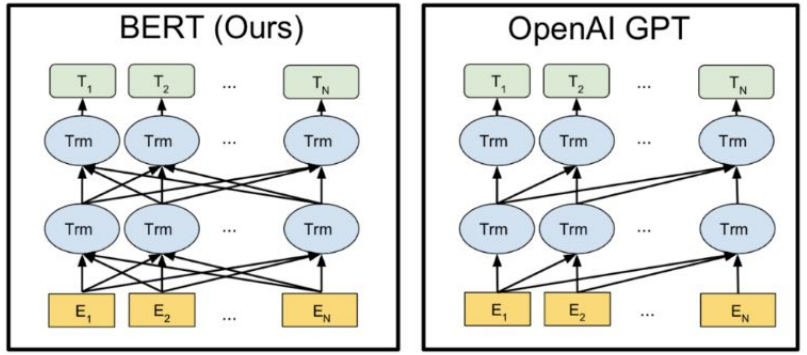
[Mikolov et al., NeurIPS-13]

[Bengio et al., JMLR-03]





[Peters et al., NAACL-18]



[Devlin et al., NAACL-19]

Unsupervised Pretraining Models...

...also rely on (local) **distributional** (co-occurrence) information. Specializing them? WIP (later)

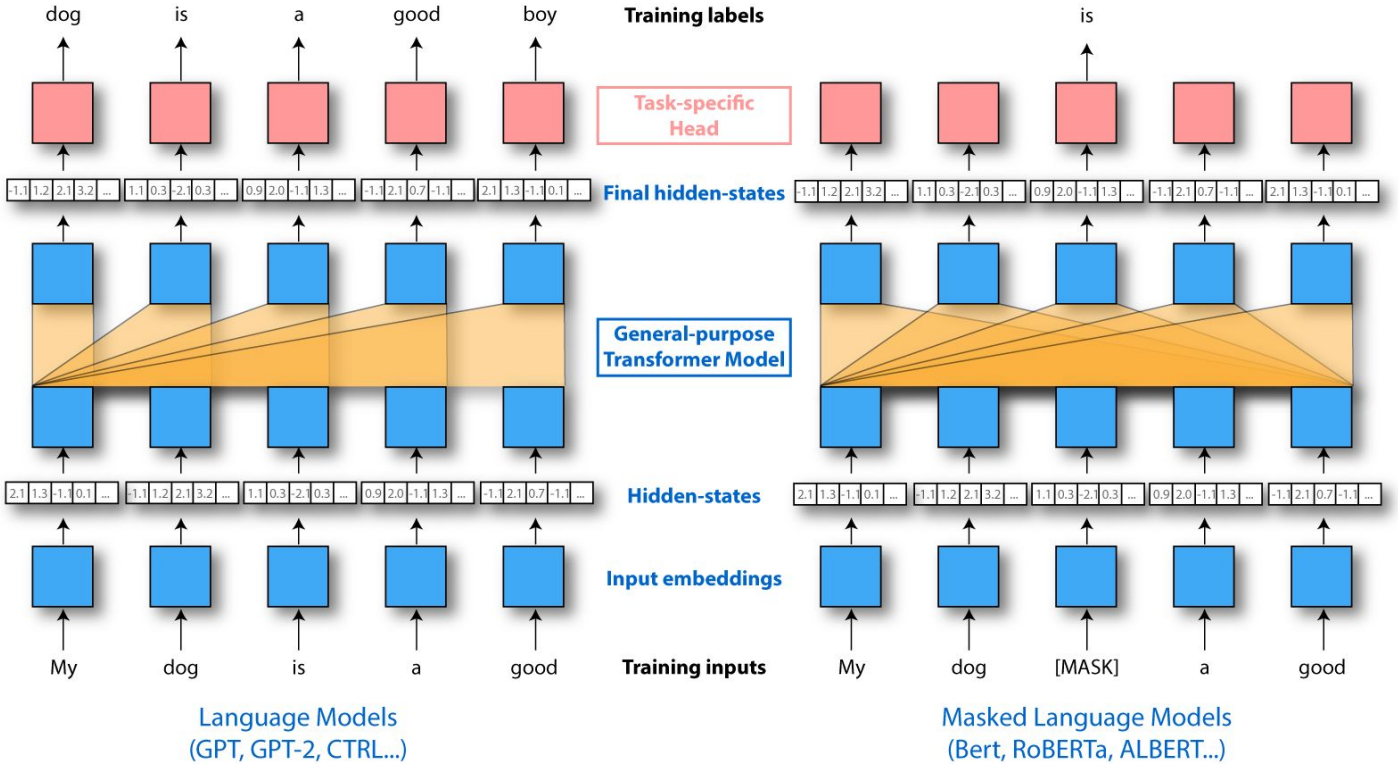
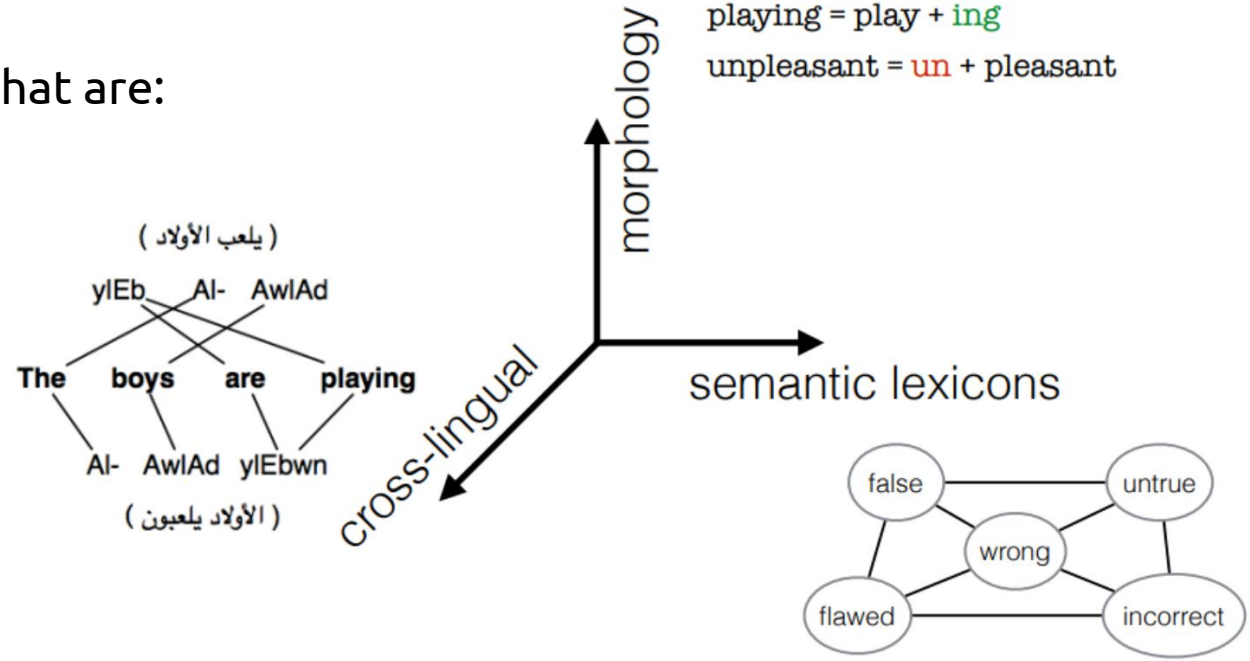


Image courtesy of Thomas Wolf (Source: Twitter)

Beyond (Limited) Word Co-Occurrence

Learning representations that are:

- richer
- more informed
- focused on a particular property (i.e., **specialized**)



(Faruqui, 2016)

What about Lexico-Semantico Resources Only?

Lexico-semantic resources clearly define and explicitly specify (meaningful) relationships between concepts:

- synonymy, antonymy, meronymy, hyponymy, co-hyponymy
- manually annotated (i.e., reliable)

Then why do we need distributional representations?

- **Limited coverage** (even in resource-rich languages): only 15.3% of words in the top 200K most frequent *fastText* vectors is available in WordNet

- Such resources exist only for a handful of languages, and are often incomplete

- Similarity measures based on paths and distances in those resources are not reliable

Context Manipulation and Unsupervised Post-Processing

(related, but not really covered in this tutorial...)

What is Context?

Representation models such as **skip-gram**, **CBOW**, or **fastText** can be trained with different contexts

Context is crucial: different contexts steer learning towards similarity or towards relatedness, or towards better representations for particular **word classes**

Investigating the **role of context**
[Melamud et al., NAACL-16; Schwartz et al., CoNLL-15, NAACL-16; Vulić et al., CoNLL-17]

Some standard context types:
1. (Ordinary) bag-of-words **(BOW)**
2. Positional **(POSIT)**
3. Dependency-based: Basic **(DEPS-NAIVE)**
4. Dependency-based: with prepositional arc collapsing **(DEPS-ARC)**

...

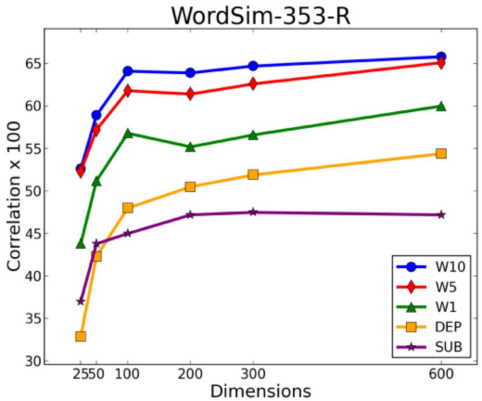
Perceptual context?

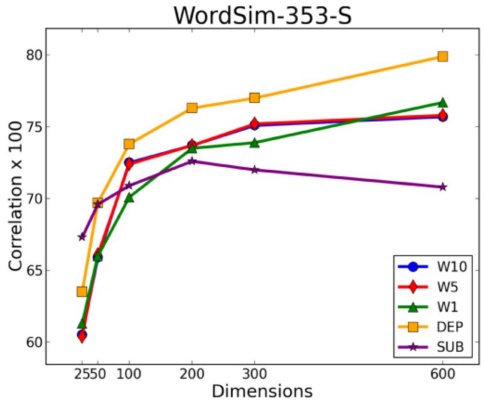
Context Impacts the Resulting Word Vectors

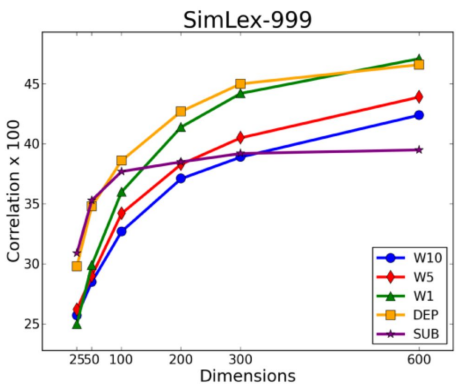
SGNS-BOW ($c = 2$)	SGNS-BOW ($c = 5$)	SGNS-DEPS
dancing	dancing	dancing
singing	singing	singing
dance	dance	rapping
dances	dances	breakdancing
breakdancing	dancers	miming
clowning	tap-dancing	busking

[Levy and Goldberg, ACL-14]

[Melamud et al., NAACL-16]







Context Impacts the Resulting Word Vectors

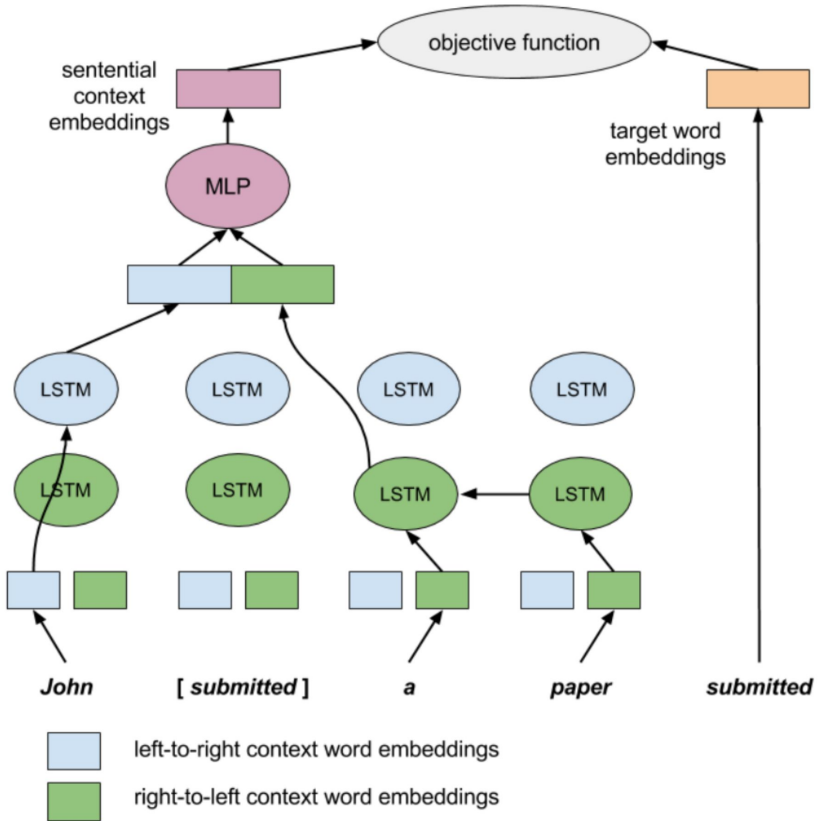
[Melamud et al., CoNLL-16]: **context2vec**

Even richer contexts -> sentential context modeled by a bidirectional LSTM

(A sort of “improved CBOW” model)

It targets functional similarity by modeling long-distance dependencies implicitly

Very good results on intrinsic semantic similarity tasks (e.g., verb similarity)



Other Context Types

Other context types:

- Attention-based CBOW (or SGNS)
  [Ling et al., EMNLP-15]
- Substitute vectors: potential filler words for the target word slot according how they 'fit' to fill the target slot; *an LM-inspired objective before it became popular (again)...*
  [Yatbaz et al., EMNLP-12; Melamud et al., NAACL-15]
- Context configuration selection for different word classes (e.g., noun-specific vs adjective-specific)
  [Schwartz et al., NAACL-16, CoNLL-17]

Bottom line: **context (type)** only defines some of the properties (implicitly!) of our initial distributional vectors

(Related: **cross-lingual grounding** and **context enrichment**)
(Related: **multi-modal/perceptual grounding** and **context enrichment**)
(Why is it related? **Adding external information** to **(monolingual)** distributional representations)

Unsupervised Post-Processing of Word Vectors

Related work, but not covered here in this tutorial *(complementary goals)*...

Can we improve distributional representations prior to injecting any knowledge?

[Mu et al., ICLR-18; Liu et al., AAAI-19; Tang et al., arXiv-19]: remove the dominating PCA components; this makes the vector space more isotropic; different flavours of the same idea

(Small) improvements on a range of word similarity/relatedness tasks and text classification

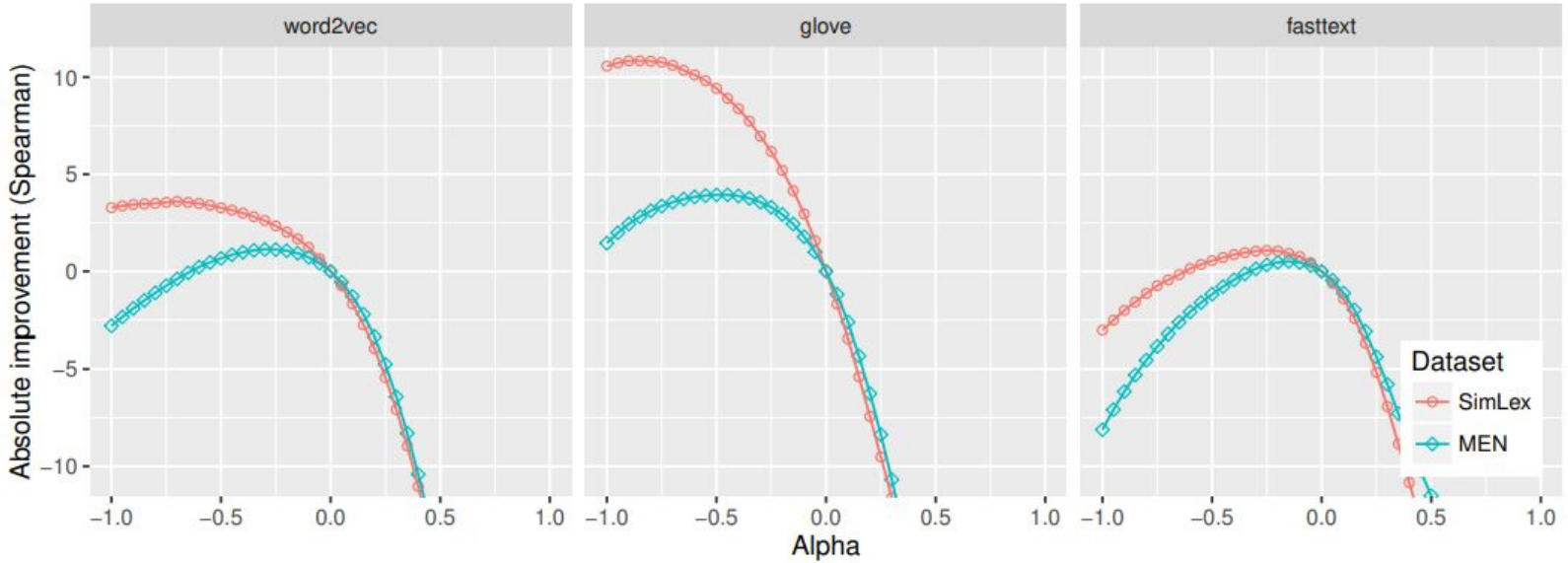
Algorithm 1: Postprocessing algorithm on word representations.

Input : Word representations $\{v(w), w \in \mathcal{V}\}$, a threshold parameter D,
Compute the mean of $\{v(w), w \in \mathcal{V}\}$, $\mu \leftarrow \frac{1}{|\mathcal{V}|}\sum_{w\in\mathcal{V}} v(w)$, $\tilde{v}(w) \leftarrow v(w) - \mu$
Compute the PCA components: $u_1, ..., u_d \leftarrow \text{PCA}(\{\tilde{v}(w), w \in \mathcal{V}\})$.
Preprocess the representations: $v'(w) \leftarrow \tilde{v}(w) - \sum_{i=1}^{D} \left(u_i^\top v(w)\right) u_i$
Output : Processed representations $v'(w)$.

Unsupervised Post-Processing of Word Vectors

[Artetxe et al., CoNLL-18]: a linear transformation that adjusts the similarity order of pretrained vectors

(Small) improvements on a range of word similarity/relatedness and analogy tasks: we can steer the embeddings towards **semantic versus syntactic** properties



Critical: These improvements *without any external knowledge* are still **very small**

A Very Short Recap

Key Idea

Refine distributional representations using external lexical resources

Why?

Distributional representations (static and contextualised alike) contain rich semantic knowledge from word co-occurrence which makes them suitable for a wide range of NLP tasks…
...but are there limits to information contained in word co-occurrence?

Key Question

Can we **specialize distributional representations** for **particular relations** (and **downstream tasks**) with the help of **available** (or **transferred** or **newly induced**) **external knowledge**?

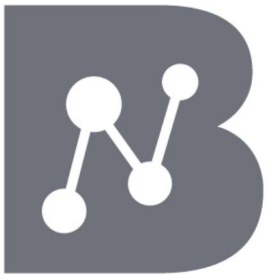
Lexical Relations and Lexical Resources

(A very short and incomplete overview…)

Enriching Distributional Models with Semantic Knowledge

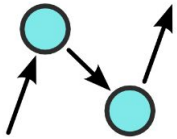
Unsupervised (i.e., self-supervised) methods making use of distributional information are both theoretically interesting and require no manual annotation.

However, if our goal is optimising **downstream performance**, why not make use of **all the lexical resources already available?**



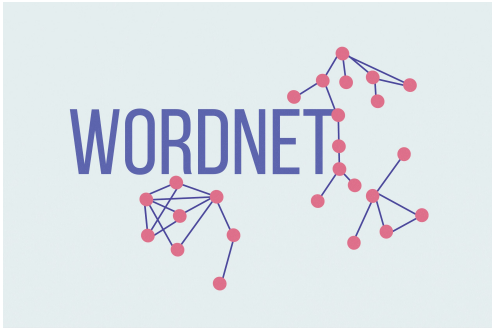
BabelNet

Paraphrase.org



ConceptNet

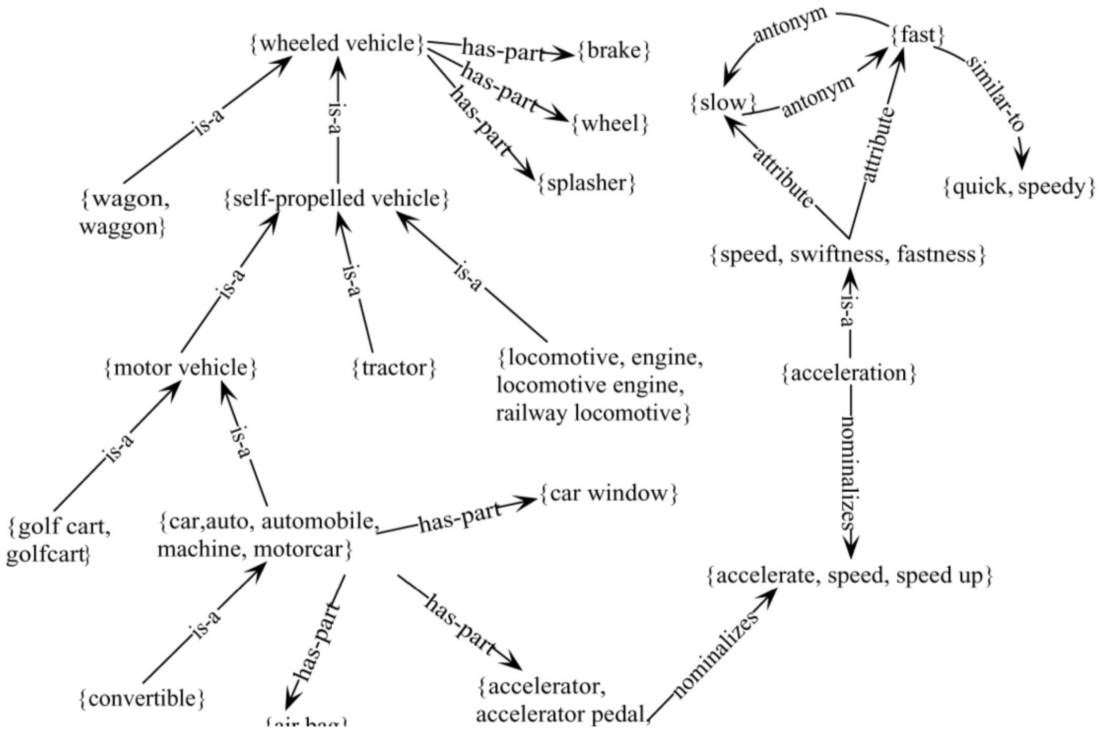
An open, multilingual knowledge graph



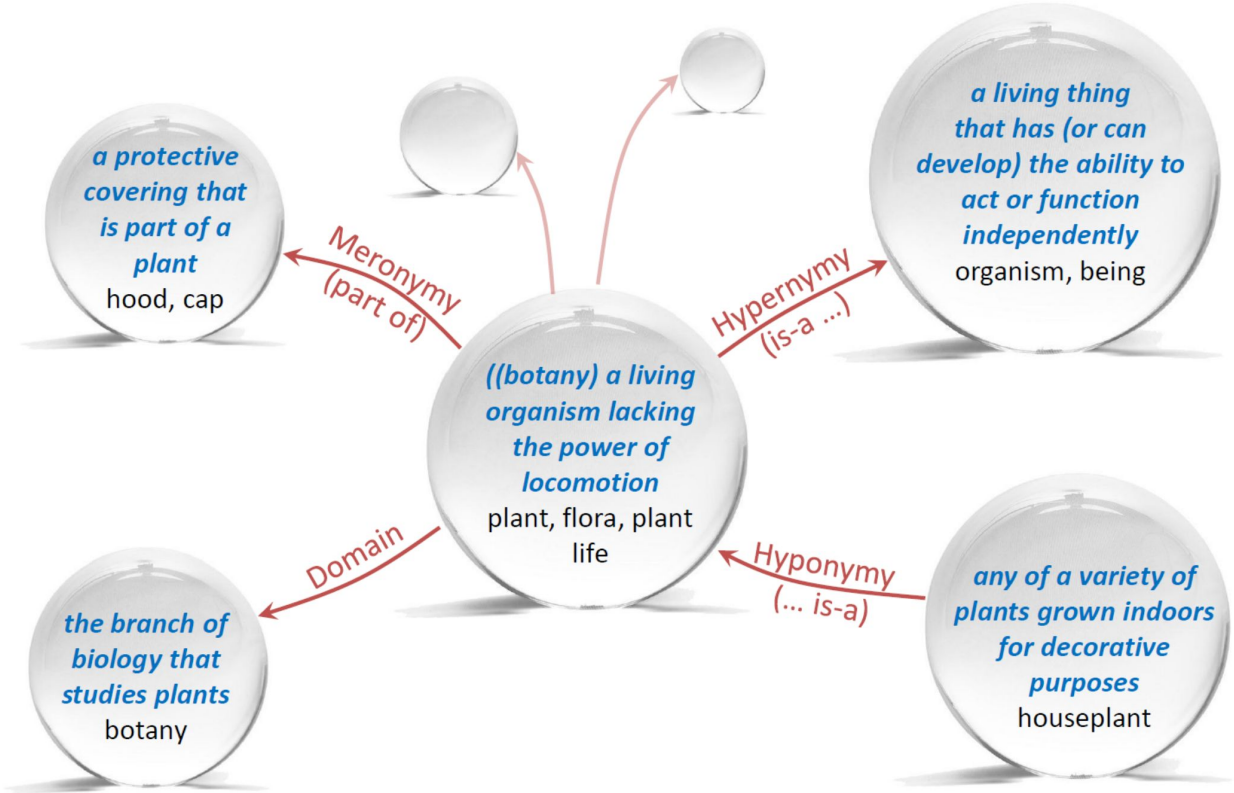
Major Lexico-Semantic Resources

WordNet [Fellbaum, 1998]

- >150K synsets
- 18 lexico-semantic relations
- Mostly unconnected POS hierarchies
  (links within POS)



WordNet: Lexical Relations



WordNet 3.0 General Statistics

Number of words, synsets, and senses

POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Polysemy information

POS	Monosemous Words and Senses	Polysemous Words	Polysemous Senses
Noun	101863	15935	44449
Verb	6277	5252	18770
Adjective	16503	4976	14399
Adverb	3748	733	1832
Totals	128391	26896	79450

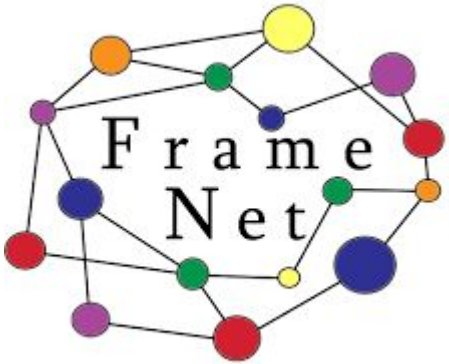
Major Lexico-Semantic Resources

Paraphrase Database (PPDB) [Ganitkevitch et al., NAACL-15; Pavlick et al., ACL-15]

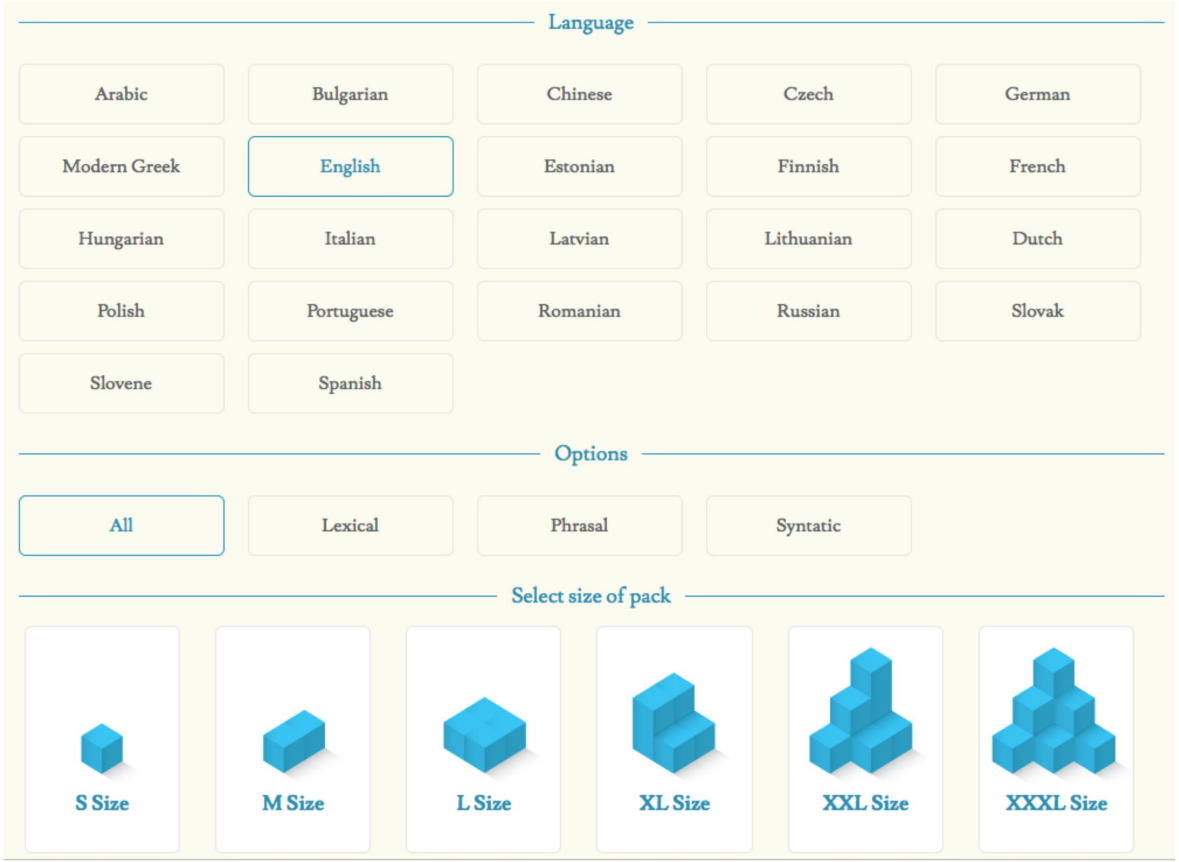
- 100M paraphrases automatically extracted paraphrases
- Entries have their own embeddings (obtained with Multiview LSA)

FrameNet [Baker et al., 1998; Ruppenhofer et al., 06, 17]

- 200K sents annotated with 1200 semantic frames
- Predicates and semantic roles of their arguments



PPDB: Some Properties



Extracted from bilingual parallel corpora through **bilingual pivoting** [Bannard and Callison-Burch, 2005]

Three types of paraphrases:

- **Lexical**
- **Phrasal**
- **Syntactic**

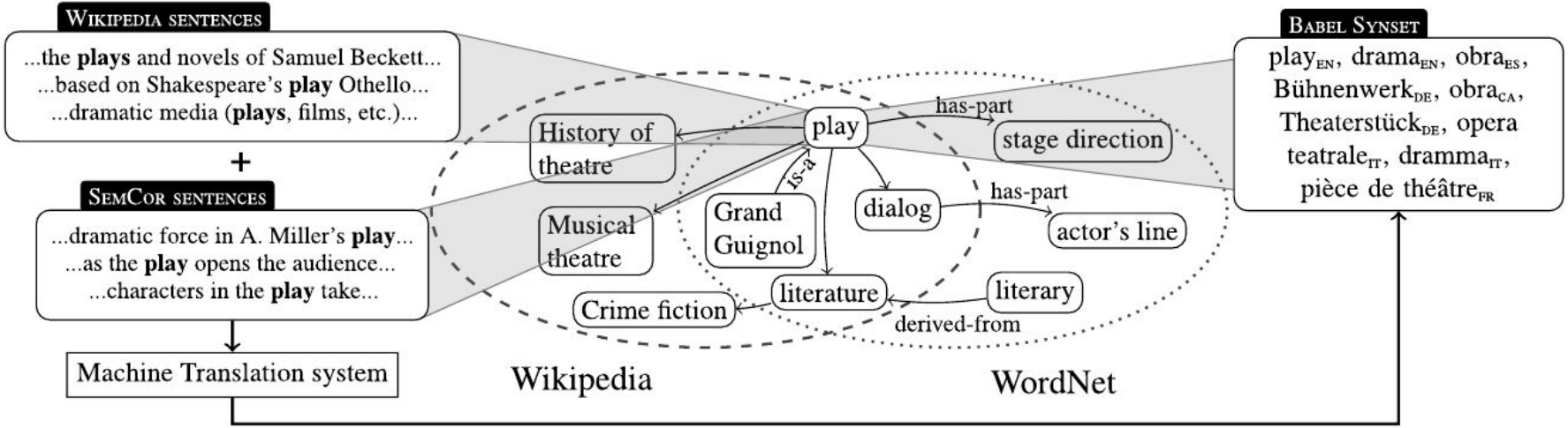
Each paraphrase is provided with a (confidence/reliability) score

PPDB2.0 contains fine-grained entailment relations, word embedding similarities, style annotations, etc.

Major Lexico-Semantic Resources

BabelNet [Navigli & Ponzetto, ACL-10; AI-12]

- Merged Wikipedia and (Multilingual) WordNet + a number of smaller resources (e.g., GeoNames, OmegaWiki, MS Terminology, VerbNet)
- 284 languages, 6M concepts, 16M synstes, 809M senses



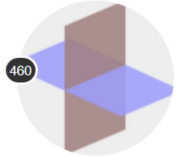
Major Lexico-Semantic Resources: BabelNet

airplane, plane, aeroplane

An aircraft that has a fixed wing and is powered by propellers or jets

ID: 00001697n | Concept

- 固定翼飛機, 飛行機, 飞龙机
- avion, aéroplane
- Flugzeug
- aereo, aeroplano, apparecchio



plane, sheet

(mathematics) an unbounded two-dimensional shape

ID: 00062766n | Concept

- 平面, 面
- plan
- Ebene (Mathematik)
- piano, piano geometrico

plane

A level of existence or development

ID: 00062767n | Concept

- 平面的存在
- plan
- Ebene
- piano, Spostamento della realtà, livello

planer, plane, planing machine

A power tool for smoothing or shaping wood

ID: 00062768n | Concept

- 刨床
- raboteuse, rabot
- Hobelmaschine
- piallatrice

plane, woodworking plane, carpenter's plane

A carpenter's hand tool with an adjustable blade for smoothing or shaping wood

ID: 00016196n | Concept

- 刨
- rabot, avion, appareil
- Hobel
- pialla, piana, pialletto

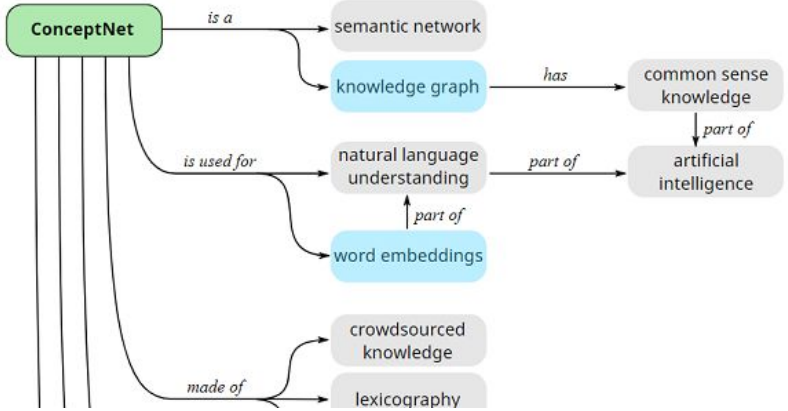
BabelNet 4.0: General Statistics

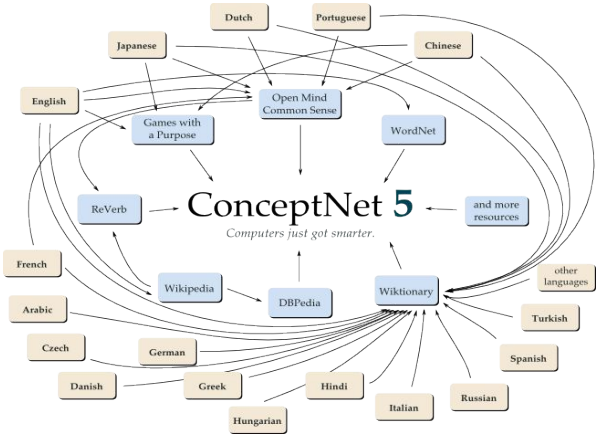
Number of languages:	284
Total number of Babel synsets:	15,780,364
Total number of Babel senses:	808,974,108
Total number of concepts:	6,113,467
Total number of Named Entities:	9,666,897
Total number of lexico-semantic relations:	277,036,611
Total number of glosses (textual definitions):	91,218,220
Total number of images:	54,229,458
Total number of Babel synsets with at least one domain:	2,637,407
Total number of Babel synsets with at least one picture:	10,522,922
Total number of sources:	47

Major Lexico-Semantic Resources

ConceptNet [Speer et al., AAAI-17]

- Common-sense lexical relations (OMCS)
- + Wiktionary and Open Multilingual WordNet
- 8M nodes, 21M edges
- 30+ rels
  - Lexico-semantic like *synonymy*
  - Commonsense like *capable-of*





Tutorial Overview

1. **Introduction and Motivation:** distributional representation models (Static vs Non-static), lexical relations, external repositories, complementarity of information
(20 minutes, Ivan)

2. **Specialisation for Semantic Similarity**: similarity vs relatedness vs other relationships, joint versus retrofitting models, explicit retrofitting versus post-specialisation, evaluating for semantic similarity
(45 minutes, Ivan)

3. **Specialisation for LE and Other Relations**: specialisation for lexical entailment, embedding hierarchies in vector spaces, explicit versus post-specialisation for LE; specialization for other relations, evaluation
(35 minutes, Goran)

4. **Cross-lingual Transfer of Specialisation**: different approaches to target language specialisation, supporting the construction of lexical resources in resource-poor language; challenges with resource-low settings
(25 minutes, Goran)

5. **Specialisation of Contextualised Representation Models**: LIBERT, K-BERT, ERNIEs, etc.
(45 minutes, Edoardo)

6. **Challenges, Open Problems, Conclusions**
(10 minutes, Edoardo)

Semantic Similarity Specialization

Specialization: Model Typology

- **Joint** specialization models
- **Retrofitting** (post-processing) models
- **Post-specialisation** models
- **Direct** or **explicit specialization** models

(True) Word Similarity

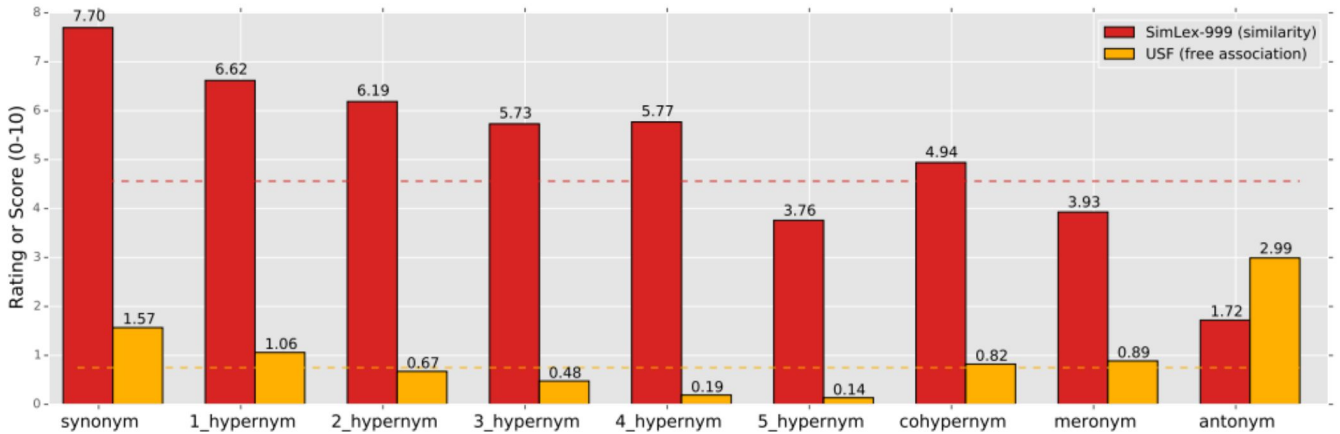
Image from:

[Hill et al., CL-15]

Synonymy is **binary** : on/off, words are synonyms or not

A looser metric: **word similarity**

Two words are **more** similar if they share **more features** of meaning. "Graded" synonymy?



Evaluation and Application? Intrinsic

Intrinsic Evaluation Resources

Most intrinsic evaluation datasets *rank* word pairs by *similarity*, *relatedness*, *lexical entailment*, or other relevant properties.

Similarity **and** Relatedness:

- RG-65 (Rubinstein and Goodenough, 1965)
- WS-353 (Finkelstein et al., 2002)
- WS-353 Sym/Rel split (Agirre et al., 2009)
- Rare Words (Luong et al., 2013)
- MEN (Bruni et al., 2014)

Evaluation and Application? Intrinsic

Lexical Entailment:

- HyperLex (Vulić et al., 2016)

Word Analogy:

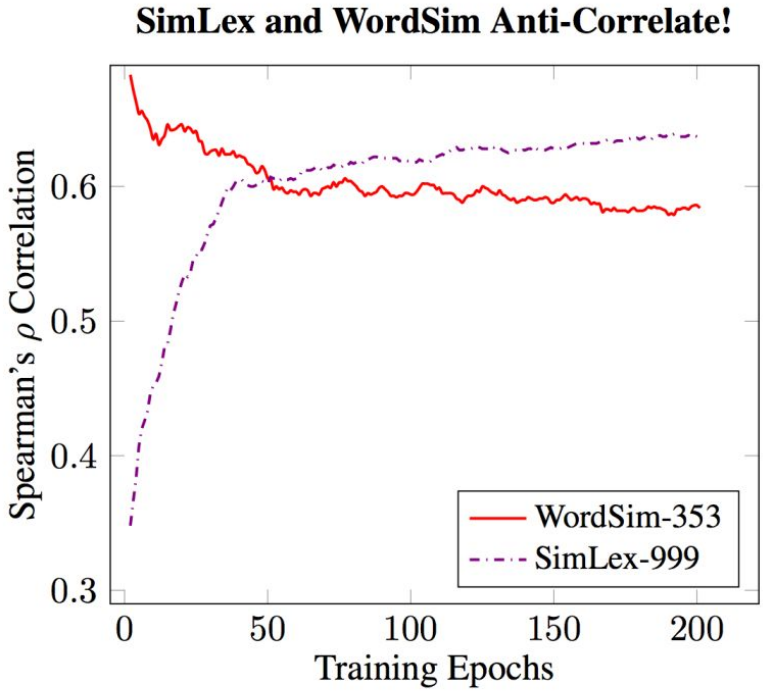
- Microsoft and Google Analogy (Mikolov et al., 2013)

Semantic Similarity (**decoupled** from Relatedness/Association):

- TOEFL Synonym Questions (Landauer and Dumais, 1997)
- GRE Antonymy (Mohammad et al., 2008)
- SimLex-999 (Hill et al., 2015)
- SimVerb (Gerz et al., 2016)
- Multilingual SimLex-999 (Leviant and Reichart, 2015)
- SemEval 2017 Task 2: Multilingual and cross-lingual semantic word similarity (Camacho-Collados et al., 2017)

Different Specialization means Different Representation

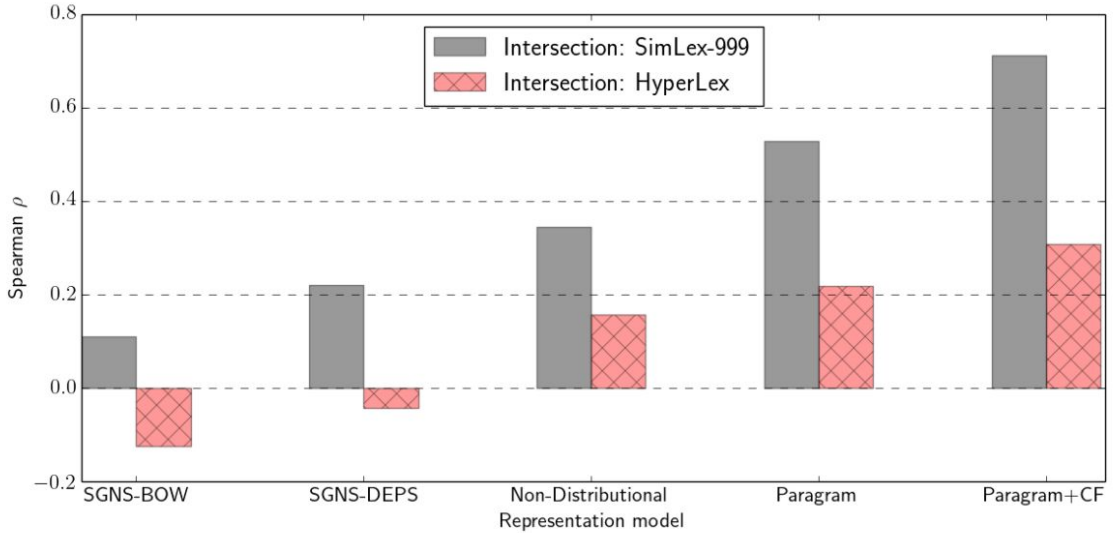
Different evaluation sets evaluate different aspects of relationship (or call it “similarity”)



Similarity $\neq$ Relatedness

Different Specialization means Different Representation

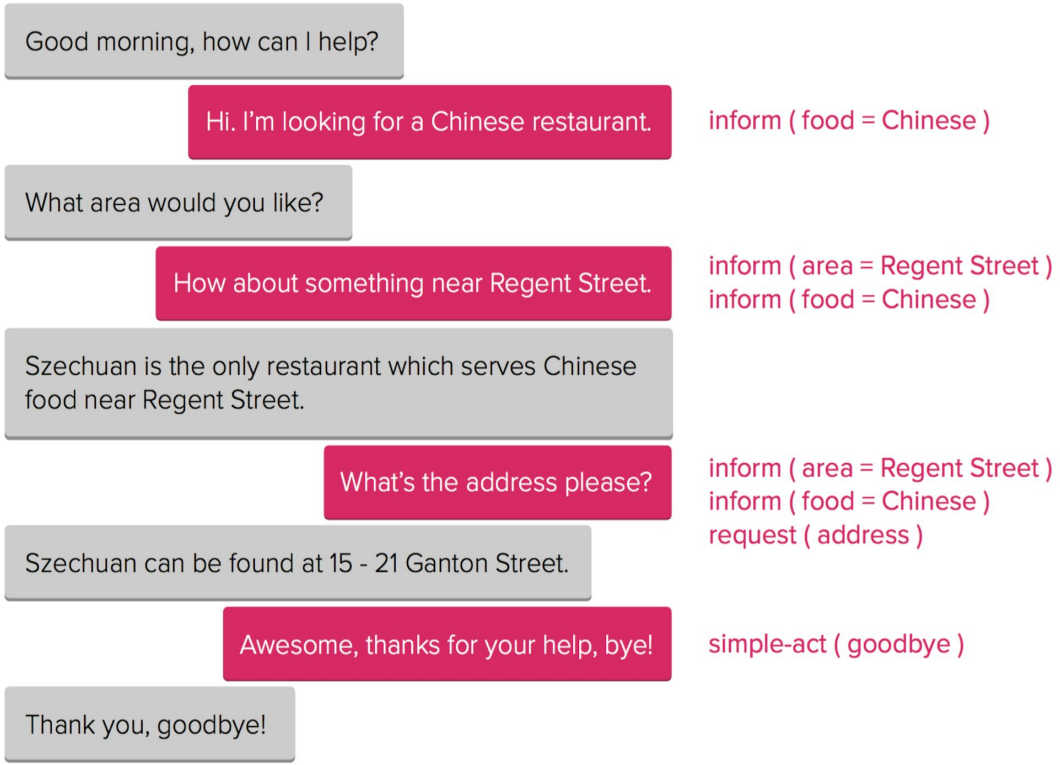
Different evaluation sets evaluate different aspects of relationship (or call it “similarity”)



Similarity ≠ Relatedness ≠ Lexical Entailment

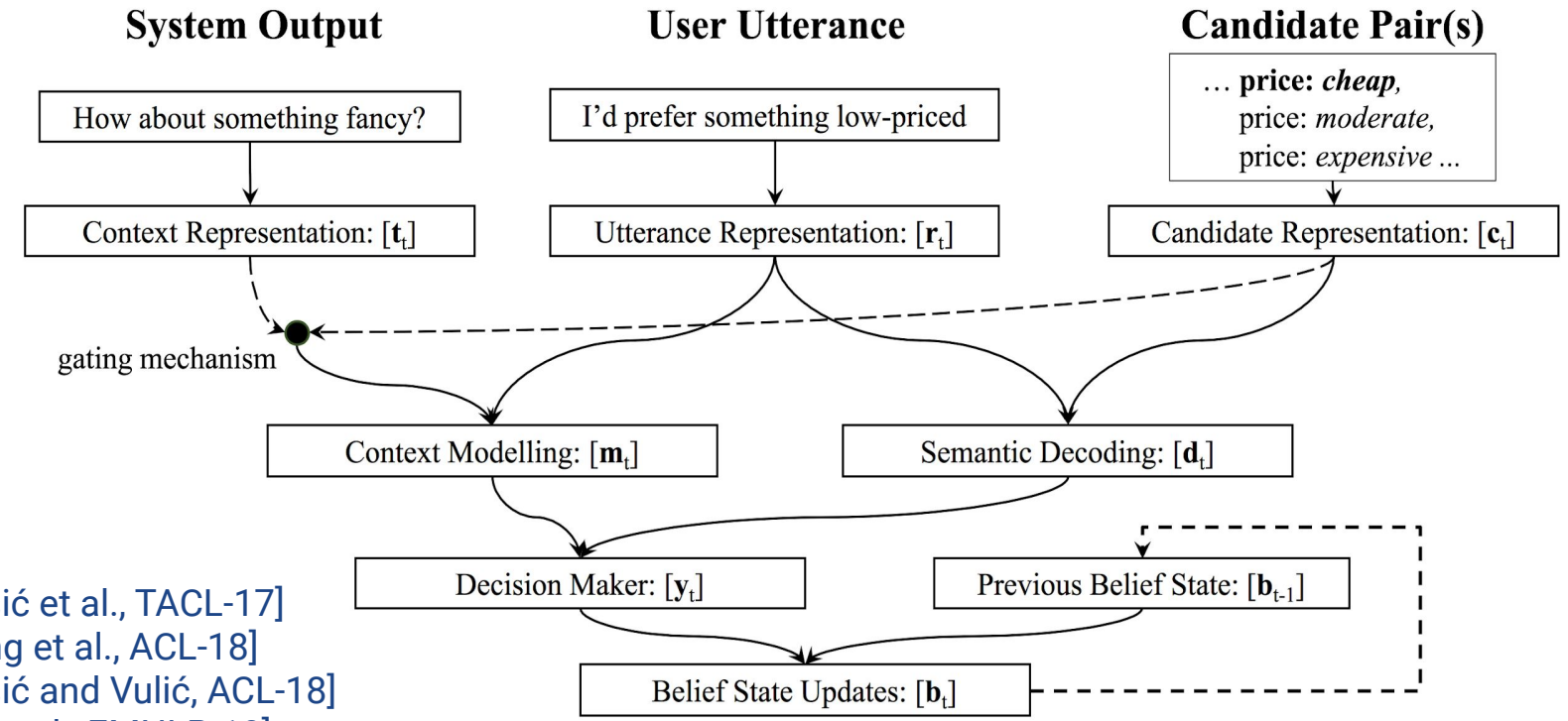
[Vulić et al., CompLing-17]

Evaluation and Application? Dialogue State Tracking



Natural Language Understanding for Dialogue (DST = Dialogue State Tracking)

Evaluation and Application? Dialogue State Tracking



[Mrkšić et al., TACL-17]
[Zhong et al., ACL-18]
[Mrkšić and Vulić, ACL-18]
[Ren et al., EMNLP-18]

Fully data-driven (neural) DST models: *distinguishing between similarity and relatedness is critical*

Why not inject the information explicitly into the model?

Evaluation and Application? Lexical Simplification

Lexical text simplification: aims to replace complex words with their simpler synonyms.

- Retaining the meaning of the original text is paramount

It is **crucial** to distinguish **similarity** from **relatedness**

Ferrari's pilot Leclerc won the race

Ferrari's driver Leclerc won the race

Ferrari's airplane Leclerc won the race

Light-LS [Glavaš and Štajner, ACL-15] is a lexical simplification tool that operates purely based on word embeddings without expensive manually simplified corpora; evaluation data set [Horn et al., ACL-14]

Joint versus Post-Processing

1. Joint "Heavy-Weight" Approaches

Induce representations *jointly*, by learning from contextual information while taking linguistic constraints into account.

2. Post-Processing Approaches

Inject semantic constraints into existing distributional vector spaces, treating them as black boxes.

Joint versus Post-Processing

Joint specialisation models

(+) Specialize the entire vocabulary (of the corpus)

(–) Tailored for a specific embedding model

Post-processing models

(–) Specialize only the vectors of words found/seen in external constraints

(+) Applicable to any pre-trained embedding space

(+) Much better performance than joint models

Linguistic Constraints Revisited

Constraint	Relation	Source
(response, reply)	SYN	WordNet
(enemy, foe)	SYN	WordNet
(wait, anticipate)	SYN	BabelNet
(doctorate, postgraduate)	SYN	BabelNet
(costs, expense)	SYN	PPDB
(miserable, poor)	SYN	PPDB
(demand, supply)	ANT	WordNet
(stand, sit)	ANT	WordNet
(worthless, valuable)	ANT	BabelNet
(commencement, finishing)	ANT	BabelNet
(dishonour, honored)	ANT	PPDB
(intellect, stupidly)	ANT	PPDB

The simple format of linguistic constraints:

It allows us to combine knowledge from diverse external sources of lexico-semantic information

Joint Specialization Models

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 0: Polarity Inducing Latent Semantic Analysis
[Yih et al., EMNLP-12]

- It combines information from the thesaurus with a regular LSA models
- Negating entries of antonyms in the word co-occurrence matrix
- Why is it important? The idea of injecting explicit knowledge into word representations existed in pre-embedding times

Joint Specialization Models

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 1: Relation Constrained Model
[Yu and Dredze, ACL-14]

It combines 1) distributional CBOW objective with 2) a set of linguistic constraints describing a relation (e.g., similarity)

$$J = \frac{1}{T}\underbrace{\sum_{t=1}^{T} \log P(w_t|w_{t-c}^{t+c})}_{\text{Distributional: CBOW}} + \underbrace{\frac{C}{N}\sum_{i=1}^{N}\sum_{w\in\mathbf{R}_{w_i}} \log P(w|w_i)}_{\text{Knowledge Resource}}$$

T = corpus size (tokens); C = interpolation weight; N = vocabulary size; c = window size; $\mathbf{R}_{w_i}$ = set of constraints (w, v) containing the word w_i as one word in a pair

Joint Specialization Models

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 2: Specializing for Relatedness or Similarity
[Kiela et al., EMNLP-15]

It combines 1) distributional SGNS objective with 2) a set of linguistic constraints describing a relation (e.g., similarity or relatedness)

Three modeling variants proposed, all similar to RCM:

1. Use all linguistic constraints as additional contexts

$$\frac{1}{T}\sum_{t=1}^{T}\Big(\underbrace{\log P(w_{t-c}^{t+c}|w_t)}_{\text{Distributional: SGNS}} + \underbrace{\sum_{w\in\mathbf{R}_{w_t}} \log P(w|w_t)}_{\text{Knowledge Resource}}\Big)$$

Joint Specialization Models

Example 2: Specializing for Relatedness or Similarity
[Kiela et al., EMNLP-15]

2. Sample one additional context for each token in corpus

$$\frac{1}{T}\sum_{t=1}^{T}\Big(\underbrace{\log P(w_{t-c}^{t+c}|w_t)}_{\text{Distributional: SGNS}} + \underbrace{[w \sim \mathcal{U}_{R_{w_t}}] \sum_{w \in \mathbf{R}_{w_t}} \log P(w|w_t)}_{\text{Knowledge Resource}}\Big)$$

3. SGNS Retrofitting: Distributional First

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{w \in \mathbf{R}_{w_t}} \log P(w|w_t)$$

$\rightarrow$ Similarity vs. relatedness: driven by different sets of constraints R

A Note on True Similarity versus Relatedness

Semantic similarity is not the only interesting semantic relation

$$J = \frac{1}{T}\underbrace{\sum_{t=1}^{T} \log P(w_t | w_{t-c}^{t+c})}_{\text{Distributional: CBOW}} + \underbrace{\frac{C}{N}\sum_{i=1}^{N}\sum_{w \in \mathbf{R}_{w_i}} \log P(w|w_i)}_{\text{Knowledge Resource}}$$

The set $\mathbf{R}_{w_i}$ may also contain pairwise constraints targeting relatedness $\rightarrow$ moving away from true similarity.

[Kiela et al., ACL 2015]: using $(cue, target)$ pairs from the USF word association data to generate pairwise "relatedness constraints"

A Note on True Similarity versus Relatedness

Method	SimLex-999	MEN
Skip-gram	0.31	0.68
Fit-Norms	0.08	0.14
Fit-Thesaurus	0.26	0.14
Joint-Norms-Sampled	0.43	**0.72**
Joint-Norms-All	0.42	0.67
Joint-Thesaurus-Sampled	0.38	0.69
Joint-Thesaurus-All	0.44	0.60
GB-Retrofit-Norms	0.32	0.71
GB-Retrofit-Thesaurus	0.38	0.68
SG-Retrofit-Norms	0.35	0.71
SG-Retrofit-Thesaurus	**0.47**	0.69

- Similarity-oriented constraints (thesaurus) are more useful for true similarity
- Not distinguishing between similarity and relatedness may be beneficial for certain applications such as **text classification, ad-hoc retrieval**, or **topic modeling**

Joint Specialization Models

Example 3: [Liu et al., ACL-15]

Linguistic knowledge is transformed into **ordinal constraints**:

$$similarity(w_i, w_j) > similarity(w_i, w_k)$$

Ordinal constraints are constructed using a selection of intuitive rules:

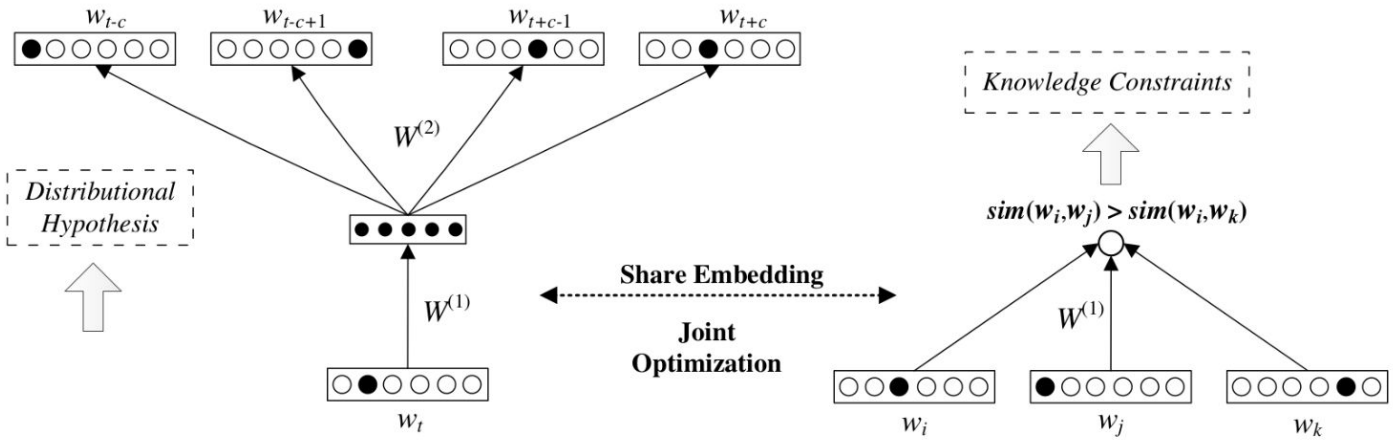
- **Synonymy/antonymy rule:** *(foolish, stupid)* $>$ *(foolish, smart)*
- **Semantic category rule:** similarities of words that belong to the same category (direct co-hyponyms) are larger than similarities of words belonging to different categories: *(mallet, plessor)* $>$ *(mallet, hacksaw)*
- **Semantic hierarchy rule:** similarities between words that have shorter distances in a semantic hierarchy should be larger than similarities of words that have longer distances: *(mallet, hammer)* $>$ *(mallet, tool)*

Joint Specialization Models

Example 3: [Liu et al., ACL-15]

Linguistic knowledge is transformed into **ordinal constraints**:

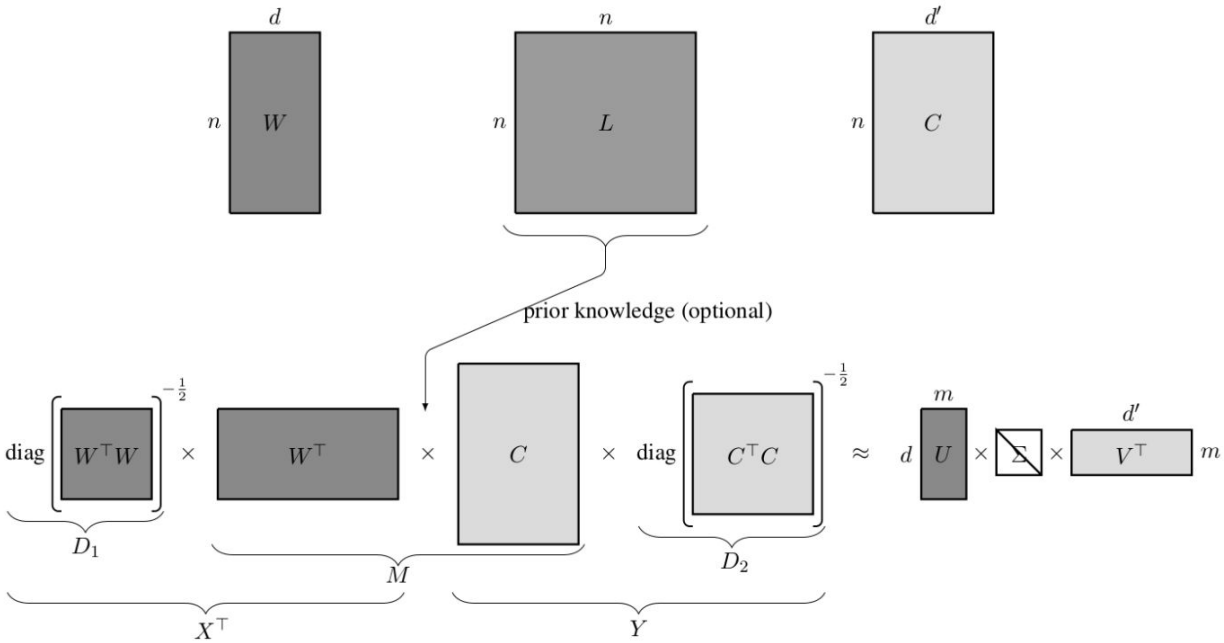
Constrained Optimisation problem using ordinal constraints



Joint Specialization Models

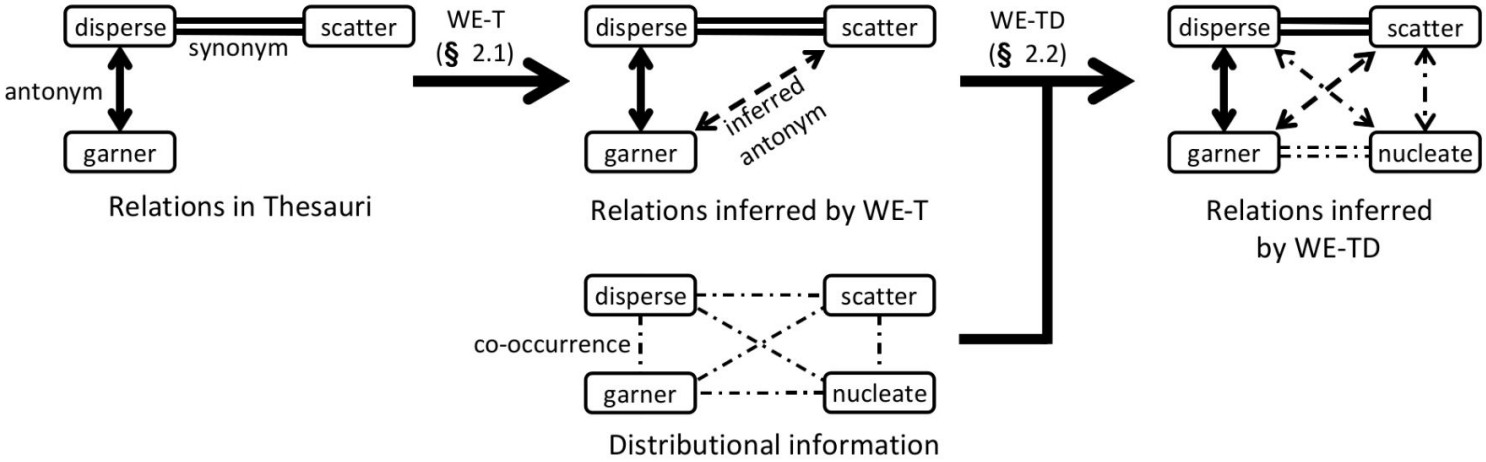
Example 4: Resource-Enriched CCA
[Osborne et al., TACL-16]

→ Two data views: pivot word view and context view



Joint Specialization Models

Example 5: specialization for antonymy detection
[Ono et al., NAACL-15]



WE-T Model: Using only external knowledge resource: i.e., embeddings of WordNet relations

WE-TD Model: Combining external knowledge with skip-gram or CBOW

Lexical Specialization vs. Definitions

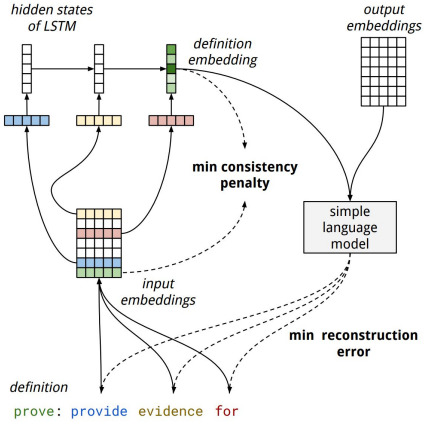
(Before we continue the specialization saga…)

A line of work that injects knowledge from **dictionary definitions** to guide the learning process [Tissier et al., EMNLP-17; Scheepers et al., WWW-18;]

- e.g., Dict2vec from [Tissier et al., EMNLP-17]**:** terminology like “strong pairs”, “weak pairs”, “positive sampling”, “controlled negative sampling”; in essence, a modification of SGNS or CBOW

car: A road vehicle, typically with four wheels, powered by an internal combustion engine and able to carry a small number of people

Another idea from [Bosc and Vincent, EMNLP-19]:
Auto-encoding dictionary definitions



Joint Specialization Models: Problems

They are “**heavy-weight”** and **demanding:**

- Training from large text corpora from scratch any time we want to change something
- Tied to the underlying distributional architecture

Long training times and **less-competitive performance**:

- Effective balancing between the two sources of information is non-trivial
- Extortionate computational complexity

Can we apply specialization as a post-processing step? **Retrofitting and pals...**

Retrofitting [Faruqui et al., NAACL-15]

First post-processing approach
Optimise a cost function which brings semantically similar words close together while keeping them (relatively) close to their initial distributional vectors.

Let V be the vocabulary (with N words), and S the set of synonymous word pairs (e.g. *sophisticated* and *refined*). Let each word pair $(x_l, x_r) \in S$ correspond to vector pairs $(\mathbf{x}_l, \mathbf{x}_r)$. The retrofitting cost function is:

$$\Psi(V, S) = \sum_{x_i \in V} \left(||\mathbf{x}_i - \widehat{\mathbf{x}_i}||^2 + \sum_{(x_i, x_j) \in S} \beta_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 \right)$$

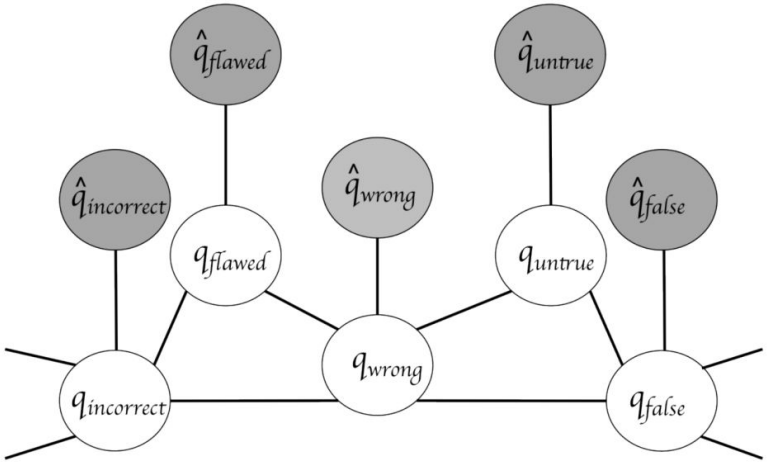
where $\beta_{i,j} = \frac{1}{deg(x_i)}$, and $deg(x_i)$ is the number of constraints in S which feature x_i. $\widehat{\mathbf{x}_i}$ is the initial distributional vector for x_i.

Retrofitting [Faruqui et al., NAACL-15]

First post-processing approach
Optimise a cost function which brings semantically similar words close together while keeping them (relatively) close to their initial distributional vectors.

$$\Psi(V, S) = \sum_{x_i \in V} \left(||\mathbf{x}_i - \widehat{\mathbf{x}_i}||^2 + \sum_{(x_i, x_j) \in S} \beta_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 \right)$$

Graph-based reinterpretation of the model: respecting “word adjacency matrices”.



Counter-Fitting [Mrkšić et al., NAACL-16]

1. **Antonym Repel (AR):**

$$\text{AR}(V') = \sum_{(u,w)\in A} ReLU\left(\delta - d(\mathbf{v}'_u, \mathbf{v}'_w)\right)$$

2. **Synonym Attract (SA):**

$$\text{SA}(V') = \sum_{(u,w)\in S} ReLU\left(d(\mathbf{v}'_u, \mathbf{v}'_w) - \gamma\right)$$

3. **Vector Space Preservation (VSP):**

$$\text{VSP}(V, V') = \sum_{i=1}^{N} \sum_{j\in N(i)} ReLU\left(d(\mathbf{v}'_i, \mathbf{v}'_j) - d(\mathbf{v}_i, \mathbf{v}_j)\right)$$

The cost function of the transformed vector space V' sums these:

$$\text{C}(V, V') = k_1 AR(V') + k_2 SA(V') + k_3 VSP(V, V')$$

PARAGRAM [Wieting et al., TACL-15]

The PARAGRAM method
It improves on retrofitting by using a more sophisticated "ATTRACT" term

If S is again the set of synonymous word pairs, the procedure iterates over mini-batches of such constraints $\mathcal{B}_S$, optimising the following cost function:

$$S(\mathcal{B}_S) = \sum_{(x_l, x_r) \in \mathcal{B}_S} \left(ReLU\left(\delta_{sim} + \mathbf{x}_l \mathbf{t}_l - \mathbf{x}_l \mathbf{x}_r\right) + ReLU\left(\delta_{sim} + \mathbf{x}_r \mathbf{t}_r - \mathbf{x}_l \mathbf{x}_r\right)\right)$$

where δ_{sim} is the similarity margin and $\mathbf{t}_l$ and $\mathbf{t}_r$ are **negative examples** for the given word pair (x_l, x_r).

PARAGRAM [Wieting et al., TACL-15]

The PARAGRAM method
It improves on retrofitting by using a more sophisticated "ATTRACT" term

If S is again the set of synonymous word pairs, the procedure iterates over mini-batches of such constraints $\mathcal{B}_S$, optimising the following cost function:

$$S(\mathcal{B}_S) = \sum_{(x_l, x_r) \in \mathcal{B}_S} \left(ReLU\left(\delta_{sim} + \mathbf{x}_l \mathbf{t}_l - \mathbf{x}_l \mathbf{x}_r\right) + ReLU\left(\delta_{sim} + \mathbf{x}_r \mathbf{t}_r - \mathbf{x}_l \mathbf{x}_r\right)\right)$$

where δ_{sim} is the similarity margin and $\mathbf{t}_l$ and $\mathbf{t}_r$ are **negative examples** for the given word pair (x_l, x_r).

PARAGRAM: The ATTRACT Term

Negative Examples for each Synonymy Pair

For each synonymy pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the one closest (cosine similarity) to $\mathbf{x}_l$ and $\mathbf{t}_r$ is closest to $\mathbf{x}_r$.

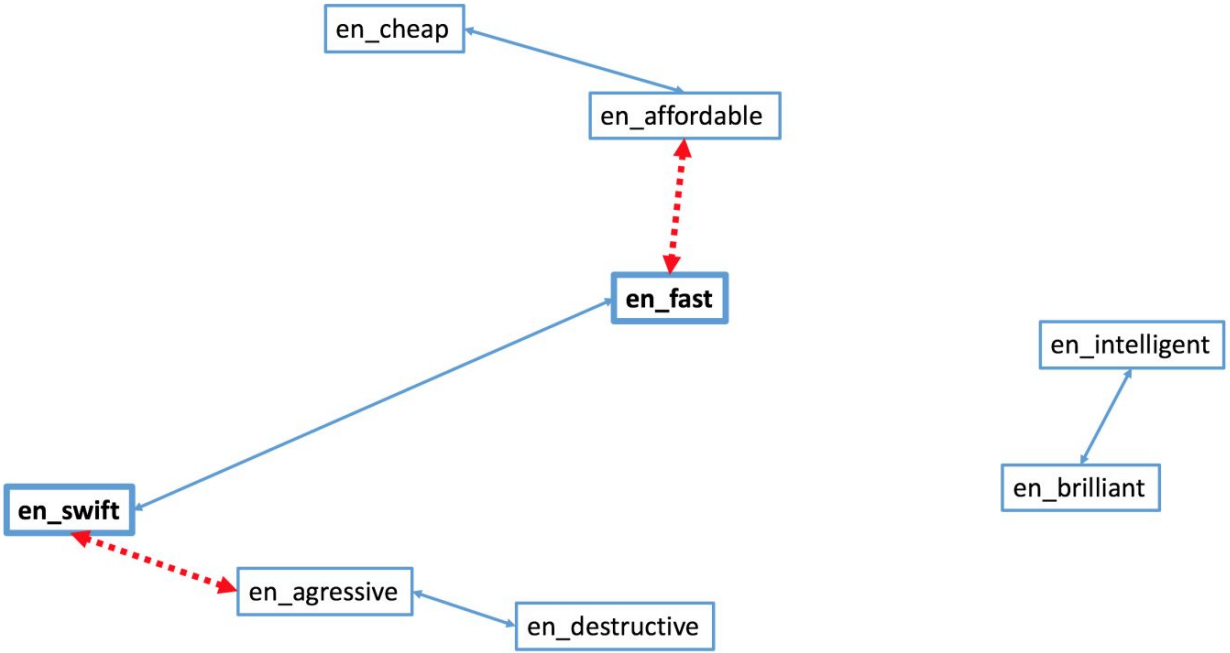
$$S(\mathcal{B}_S) = \sum_{(x_l, x_r) \in \mathcal{B}_S} \left(ReLU\left(\delta_{sim} + \mathbf{x}_l\mathbf{t}_l - \mathbf{x}_l\mathbf{x}_r\right) + ReLU\left(\delta_{sim} + \mathbf{x}_r\mathbf{t}_r - \mathbf{x}_l\mathbf{x}_r\right)\right)$$

The two negative examples are used to force synonymous pairs to be closer to each other than to their respective negative examples (i.e. to any of the remaining words in the current mini-batch).

PARAGRAM: The ATTRACT Term

Negative Examples for each Synonymy Pair

For each synonymy pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the one closest (cosine similarity) to $\mathbf{x}_l$ and $\mathbf{t}_r$ is closest to $\mathbf{x}_r$.



Exactly the same idea (only at the sentence level) was used to inform NMT training.

[Wieting et al., ACL-19]

PARAGRAM: Regularization

Semantic preservation: do not forget useful (and rich) distributional knowledge coded in the vector space

L2 Regularisation

$$R(V) = \sum_{x_i \in V} \lambda_{reg} \left\| \widehat{\mathbf{x}_i} - \mathbf{x}_i \right\|_2$$

This term is near-identical to the one in retrofitting: the *lambda* hyper-parameter can be fine-tuned to "balance" between external knowledge and distributional knowledge

Preserving distributional "relations" only if they do not contradict the injected linguistic constraints.

The ATTRACT-REPEL Model

The **Attract-Repel** model [Mrkšić et al., TACL-17] extends the PARAGRAM model with an additional **Repel** term.

The Repel Term

$$A(\mathcal{B}_A) = \sum_{(x_l, x_r) \in \mathcal{B}_A} \left(ReLU\left(\delta_{rpl} + \mathbf{x}_l \mathbf{x}_r - \mathbf{x}_l \mathbf{t}_r\right) + ReLU\left(\delta_{rpl} + \mathbf{x}_l \mathbf{x}_r - \mathbf{x}_r \mathbf{t}_r\right)\right)$$

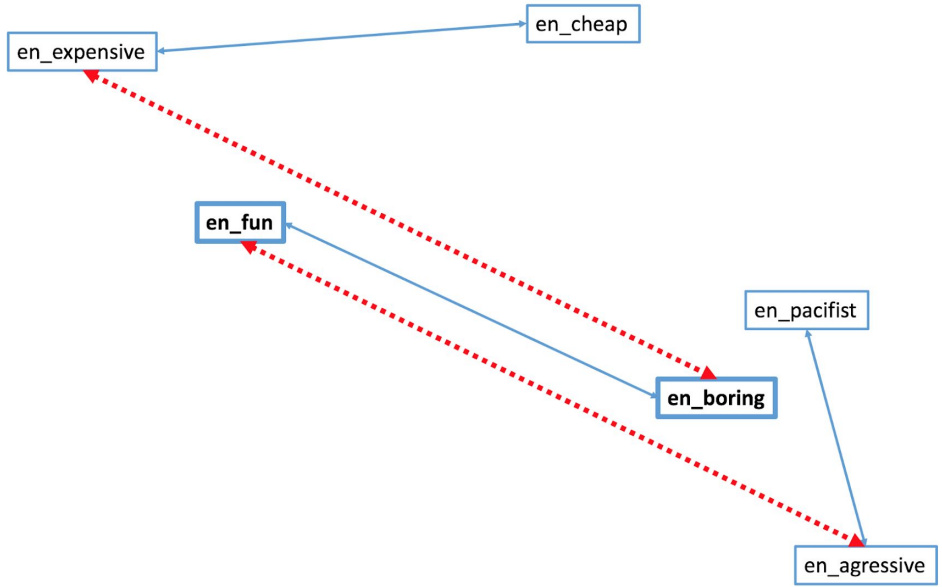
The **repel** term pushes words in undesirable relations (such as antonymy) away from each other in the reshaped vector space.

These constraints can be monolingual (e.g., *en_brave* and *en_timid*) or cross-lingual (*en_peace* and *fr_guerre*).

Repel Term: Negative Examples

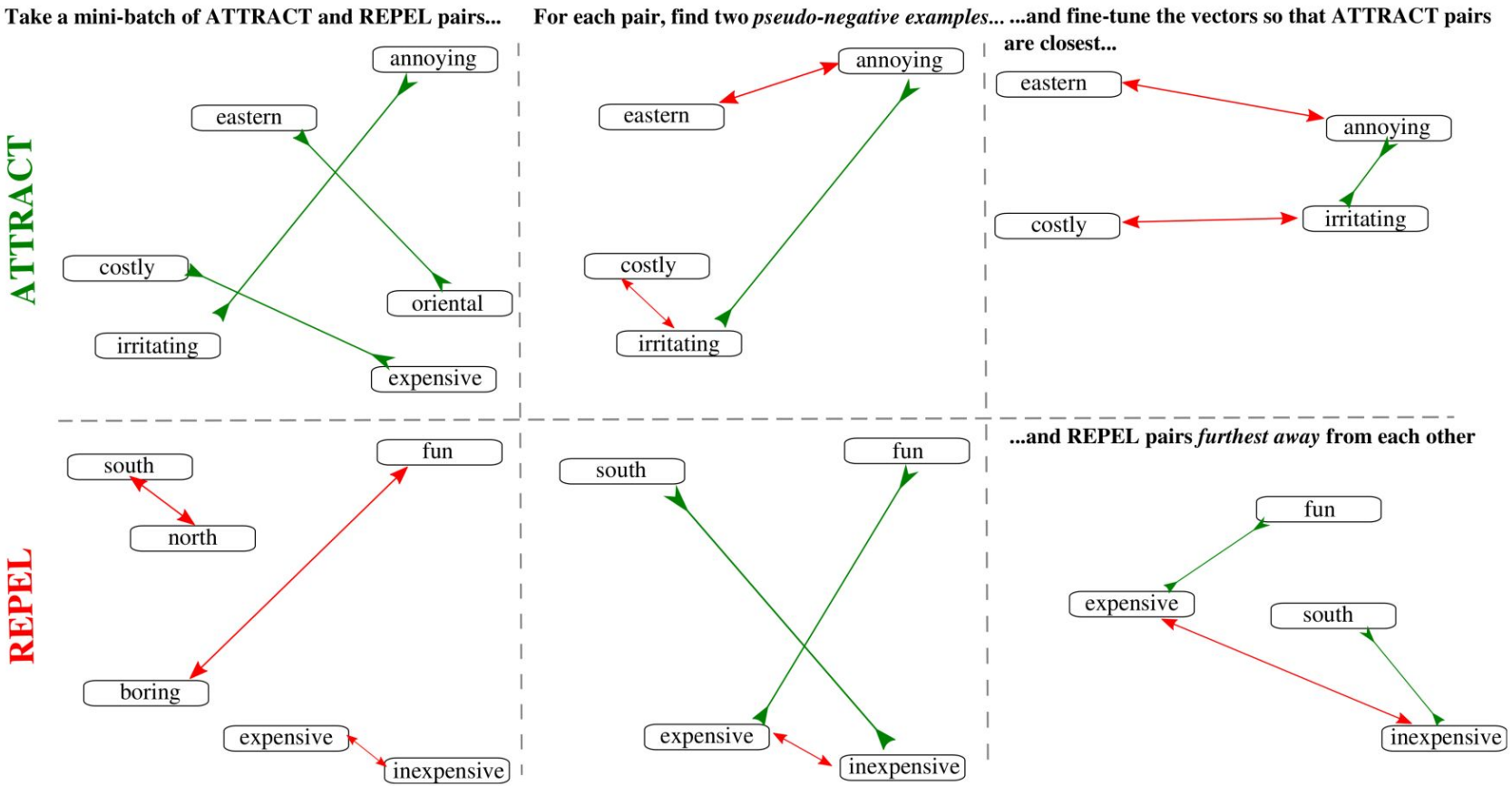
Negative Examples for each Antonymy Pair

For each antonymy pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the one furthest away from $\mathbf{x}_l$ and $\mathbf{t}_r$ is the one furthest from $\mathbf{x}_r$.



The idea of how to choose the negative examples is analogous to the Attract term

Attract-Repel in a Nutshell



How are Attract-Repel and Counter-Fitting Different?

$$S(\mathcal{B}_S) = \sum_{(x_l,x_r)\in\mathcal{B}_S} [\,\tau\,(\delta_{syn} + \mathbf{x}_l\mathbf{t}_l - \mathbf{x}_l\mathbf{x}_r) + \tau\,(\delta_{syn} + \mathbf{x}_r\mathbf{t}_r - \mathbf{x}_l\mathbf{x}_r)\,]$$

ATTRACT loss

$$A(\mathcal{B}_A) = \sum_{(x_l,x_r)\in\mathcal{B}_A} [\,\tau\,(\delta_{ant} + \mathbf{x}_l\mathbf{x}_r - \mathbf{x}_l\mathbf{t}_l) + \tau\,(\delta_{ant} + \mathbf{x}_l\mathbf{x}_r - \mathbf{x}_r\mathbf{t}_r)\,]$$

REPEL loss

$$R(\mathcal{B}_S,\mathcal{B}_A) = \sum_{\mathbf{x}_i\in V(\mathcal{B}_S\cup\mathcal{B}_A)} \lambda_{reg}\,\|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|_2$$

REGULARIZATION loss

Attract-Repel: the final objective

1. **Context-Sensitive Updates with AR**
   - Affecting both the positive pair and the negative examples
2. **Regularization**
   - Opaque and slow procedure with counter-fitting; simple L2-regularization of AR just works better

The high-level idea is indeed the same (and quite generic).

Atrract-Repel is the best performing specialization model according to a recent large empirical study [Lastra-Diaz et al., 2019]

Linguistic Constraints Steer the Specialization Process

From a very general perspective: ATTRACT and REPEL constraints

We can experiment with a variety of **constraint configurations**

`syn` (AC)	`hyp1` (AC)	`antexp` (RC)
(outburst, outbreak)	(discordance, dissonance)	(smooth, shake)
(safe, secure)	(postmen, deliverymen)	(clear, obscurity)
(cordial, warmhearted)	(employee, worker)	(relief, pressure)
(answer, response)	(swap, exchange)	(half, full)

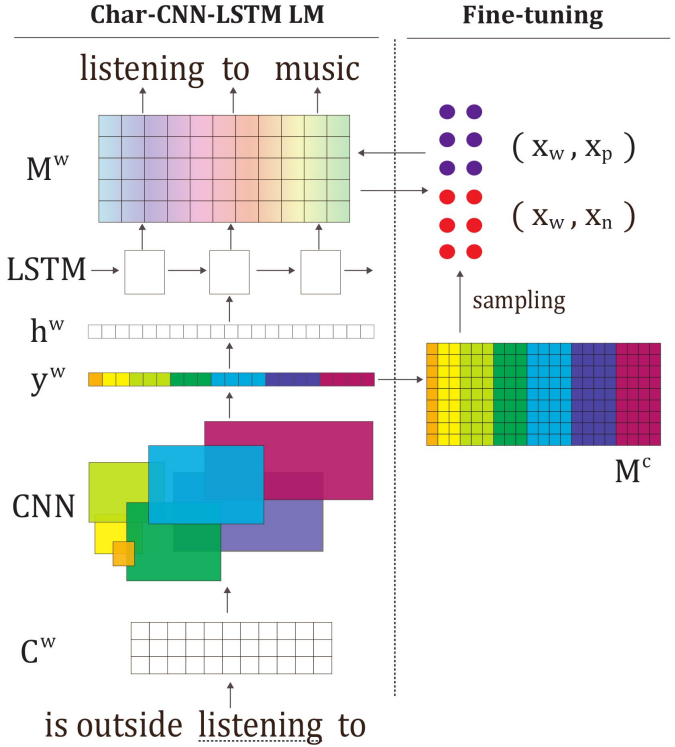
Linguistic Constraints Steer the Specialization Process

Model	SimLex	SimVerb
SGNS-GN [Mikolov et al., NIPS 2013]	0.414	0.348
Symmetric Patterns [Schwartz et al., CoNLL 2015]	0.563	0.328
Non-distributional [Faruqui et al., ACL 2015]	0.578	0.596
Joint Specialisation [Nguyen et al., ACL 2016]	0.590	0.516
Paragram-SL999 [Wieting et al., TACL 2016]	0.690	0.540
Counter-fitting [Mrkšić et al., NAACL 2016]	0.740	0.628
AR: BabelNet [Mrkšić et al., TACL 2017]	0.751	0.674
RC: `ant`	0.596	0.589
RC: `antexp`	0.606	0.551
AC: `syn`	0.748	0.728
AC: `hyp1`	0.546	0.387
AC: `syn`, RC: `ant`	0.778	0.767
AC: `syn`, RC: `antexp`	0.736	0.708
AC: `syn+hyp1`, RC: `ant`	**0.791**	**0.770**
AC: `syn+hyp1`, RC: `antexp`	0.751	0.710
Mean inter-annotator agreement	0.779	0.864

[Vulić and Korhonen, RepL4NLP-18]

Linguistic Constraints Steer the Specialization Process

An unexpected application: **smoothing the output matrix** for **language modeling in morphologically-rich languages**; relying on subword-level similarity of lexical items to mitigate data sparsity



[Gerz et al., TACL-18]

A post-processing specialisation model integrated into LM training

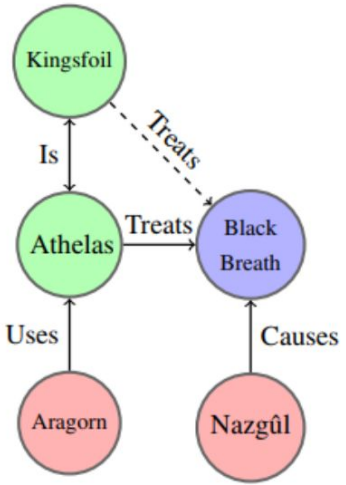
Improvements in perplexity scores on 47/50 diverse languages

	Word	Nearest Neighbours
DE	Ursprünglichkeit	**ursprünglich**e, **Ur**stoff, **ursprünglich**en
	Mittelwert	**Mittelwert**en,Regel**werk**es,**Mittel**weser
	effektiv	**Effekt**,Per**fekt**,**Effekt**e,per**fekt**en,Res**pekt**
JA	大学	大金, 大石, 大震災, 大空, 大野
	ハイク	ハイム , バイク, メイク, ハッサク
	1725	**1825**, **1625**, **1524**mm, **1728**
	Magenta	**Ma**plet, **Ma**ya, **Ma**nagement

Functional Retrofitting

Functional Retrofitting: a recent extension/generalisation of the key idea: explicitly modeling pairwise relations

Tailored for
KB completion tasks



[Lengerich et al., COLING-18]
https://github.com/roaminsight/roamresearch

Similar behavior achieved by multiple function-specific Attract-Repel models

Functional Retrofitting

Original retrofitting objective

$$\Psi_{\mathcal{G}}(\mathcal{Q}) = \sum_{i \in \mathcal{V}} \alpha_i \left\| \boldsymbol{q_i} - \hat{\boldsymbol{q}}_{\boldsymbol{i}} \right\|^2 + \sum_{(i,j,r) \in \mathcal{E}} \beta_{ij} \left\| \boldsymbol{q_i} - \boldsymbol{q_j} \right\|^2$$

Functional/generalised retrofitting objective

$$\Psi_{\mathcal{G}}(\mathcal{Q}; \mathcal{F}) = \sum_{i \in \mathcal{Q}} \alpha_i ||\boldsymbol{q_i} - \hat{\boldsymbol{q}}_{\boldsymbol{i}}||^2 + \sum_{(i,j,r) \in \mathcal{E}} \beta_{i,j,r} f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) - \sum_{(i,j,r) \in \mathcal{E}^-} \beta_{i,j,r} f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) + \sum_{r \in \mathcal{R}} \rho_\lambda(f_r)$$

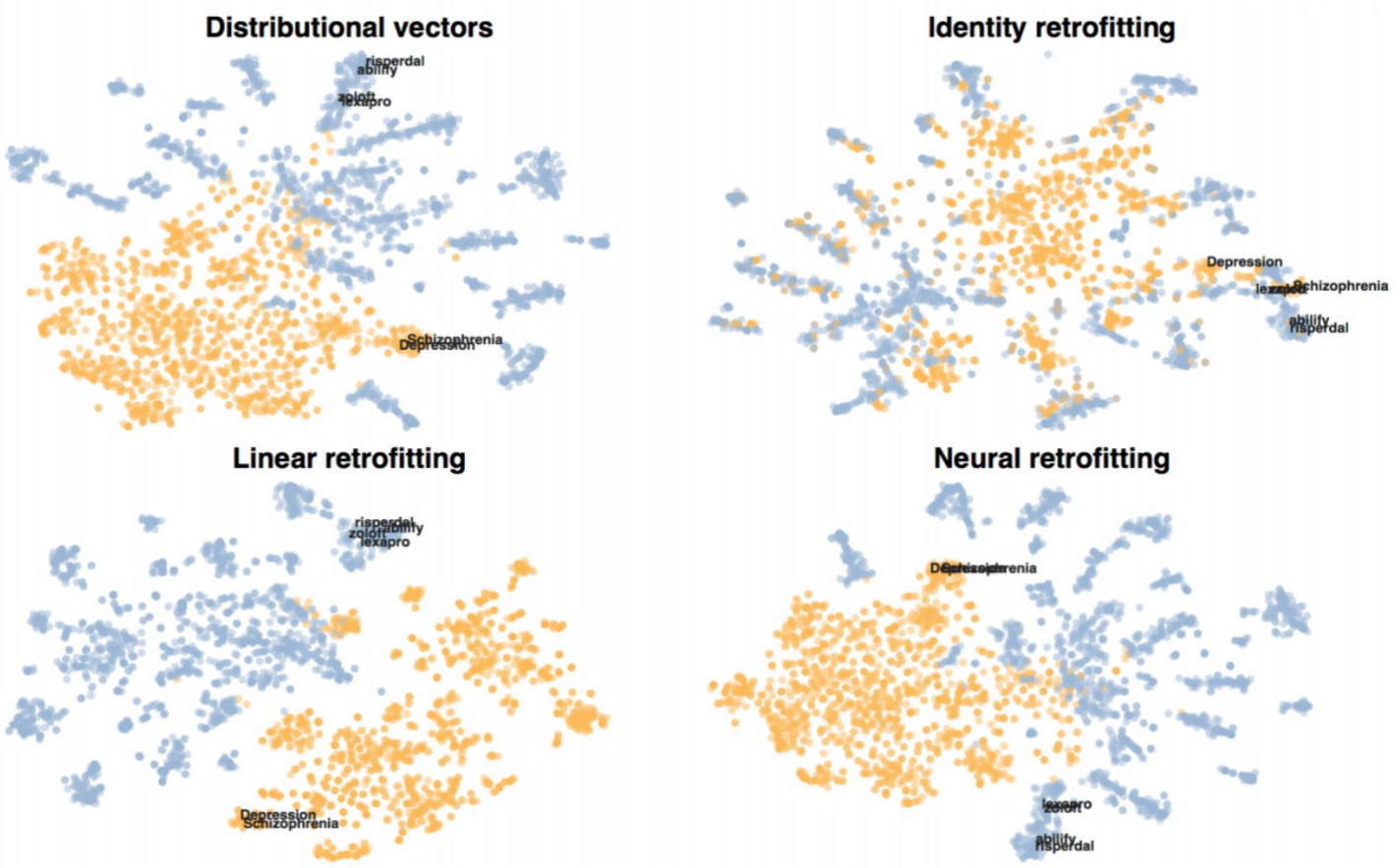
$f_r(\mathbf{q_i}, \mathbf{q_j})$ is a **relational penalty function**

Standard functions from the literature: TransE, TransH, TransR,...

In general: $f_r(\mathbf{q_i}, \mathbf{q_j}) = ||g_r(\mathbf{q_i}) + \mathbf{b}_r - h_r(\mathbf{q_j})||_2$

[Lengerich et al., COLING-18]

Functional Retrofitting



Back to Linguistic Constraints

Post-processing models (e.g., Attract-Repel) are quite **versatile:**

- we are not limited to monolingual constraints only, and can also work with **multilingual** and **cross-lingual** constraints at the same time

Constraint	Relation	Source
(response, reply)	SYN	WordNet
(enemy, foe)	SYN	WordNet
(wait, anticipate)	SYN	BabelNet
(doctorate, postgraduate)	SYN	BabelNet
(costs, expense)	SYN	PPDB
(miserable, poor)	SYN	PPDB
(demand, supply)	ANT	WordNet
(stand, sit)	ANT	WordNet
(worthless, valuable)	ANT	BabelNet
(commencement, finishing)	ANT	BabelNet
(dishonour, honored)	ANT	PPDB
(intellect, stupidly)	ANT	PPDB

We can use **BabelNet,** (**PanLex**) dictionaries, multilingual WordNets, ConceptNet

(even at the same time)

Which Constraints are Useful for Similarity Specialization?

Word Vectors	English	German	Italian	Russian
Monolingual Distributional Vectors	0.32	0.28	0.36	0.38
Attract-Repel: Mono-Syn	0.56	0.40	0.46	0.53
Attract-Repel: Mono-Ant	0.42	0.30	0.45	0.41
Attract-Repel: Mono-Syn + Mono-Ant	0.65	0.43	0.56	0.56
Attract-Repel: Cross-Syn	0.57	0.53	0.58	0.46
Attract-Repel: Mono-Syn + Cross-Syn	0.61	0.58	0.59	0.54
Attract-Repel: All Constraints	**0.70**	**0.62**	**0.68**	**0.61**

Multilingual SimLex-999 performance of EN-DE-IT-RU vectors

Again, specialization can be seen as a guided reflection of the injected constraints…

Cross-lingual constraints are useful: implicit disambiguation on both sides...

How Important is the Starting Distributional Space?

Two Sources of Information

Final vectors combine initial distributional vectors with external knowledge. Which one is more important?

Word Vectors	EN	DE	IT	RU
Random Vectors (No Information)	0.01	-0.03	0.02	-0.03
A-R: Monolingual Cons.	0.54	0.33	0.29	0.35
A-R: Mono + Cross-Ling.	0.66	0.49	0.59	0.51
Distributional Wiki Vectors	0.32	0.31	0.28	0.19
A-R: Monolingual Cons.	0.61	0.48	0.53	0.52
A-R: Mono + Cross-Ling.	0.66	0.60	0.65	0.54
A-R: Non-Wiki + Mono + Cross	0.70	0.62	0.68	0.61

Nota bene: SimLex-style evaluations can deceive; (almost) everything is **seen** in the external resource

Post-Processing Methods: Problems

Joint specialisation models

(+) Specialize the entire vocabulary (of the corpus)

(–) Tailored for a specific embedding model

Post-processing models

(–) Specialize only the vectors of words found/seen in external constraints

(+) Applicable to any pre-trained embedding space

(+) Much better performance than joint models

Full Specialization: Requirements

Best of both worlds?

- **Performance** and **flexibility** of retrofitting models
- while **specializing the entire vocabulary** (full-vocabulary specialization)

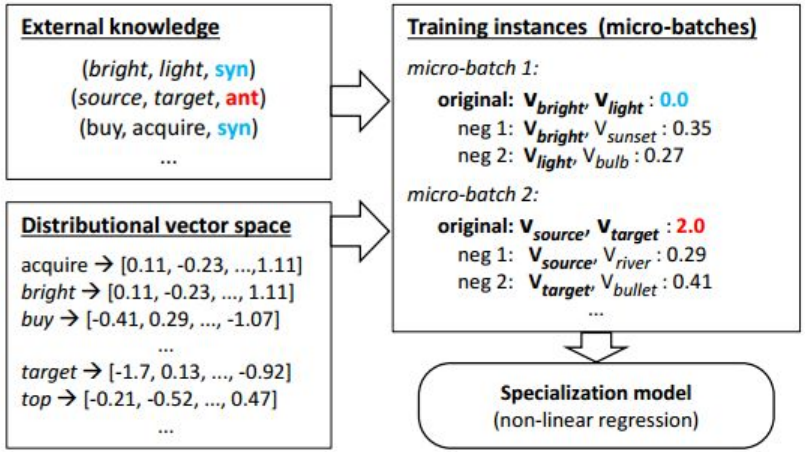
First idea: Direct or **explicit retrofitting**

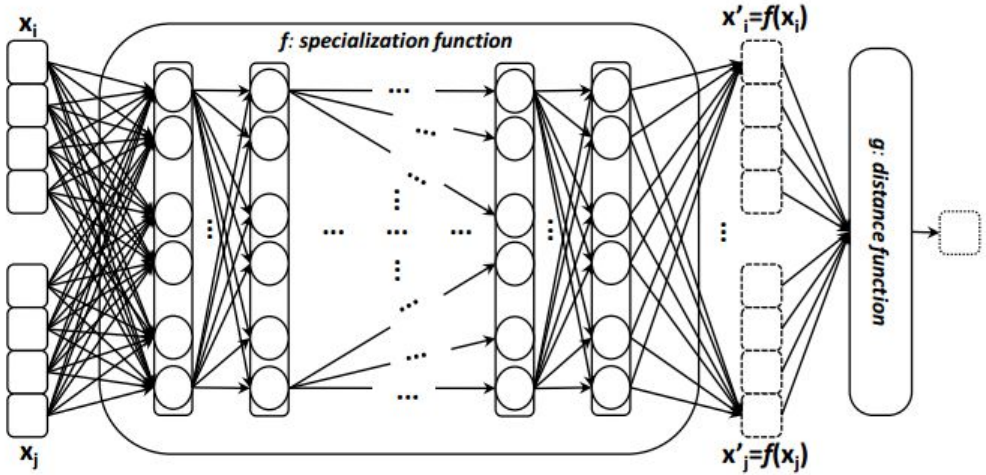
- Learn a global (explicit) retrofitting function
- External linguistic constraints are used as **training examples**

[Glavaš and Vulić, ACL-18]

https://github.com/codogogo/explirefit

Explicit Retrofitting [Glavaš and Vulić, ACL-18]





Constraints (synonyms and antonyms) used directly as training examples to learn an **explicit (global) specialization function**

The specialization function is **non-linear** (deep non-linear FFN)

Explicit Retrofitting [Glavaš and Vulić, ACL-18]

Similar to **Attract-Repel**, but framing it as “training a global function”...

- Specialization function: $\mathbf{x}' = f(\mathbf{x})$
- Distance function: $g(\mathbf{x}_1, \mathbf{x}_2)$
- Assumptions
  1. (w_i, w_j, syn) – embeddings as close as possible after specialization
     $$g(\mathbf{x}_i', \mathbf{x}_j') = g_{min}$$
  2. (w_i, w_j, ant) – embeddings as far as possible after specialization
     $$g(\mathbf{x}_i', \mathbf{x}_j') = g_{max}$$
  3. (w_i, w_j) – the non-costraint words stay at the same distance
     $$g(\mathbf{x}_i', \mathbf{x}_j') = g(\mathbf{x}_i, \mathbf{x}_j)$$

Explicit Retrofitting [Glavaš and Vulić, ACL-18]

Similar to **Attract-Repel**, but framing it as “training a global function”...

- **Contrastive Objective (CNT)**

$$J_{CNT} = \sum_{M_s \in S} \sum_{i=2}^{2K+1} \left(\underbrace{(g^i - g_{min})}_{\text{„Gold” diff.}} - \underbrace{(g'^i - g'^1)}_{\text{Predicted diff.}} \right)^2 + \sum_{M_a \in A} \sum_{i=2}^{2K+1} \left((g_{max} - g^i) - (g'^1 - g'^i) \right)^2$$

= 0

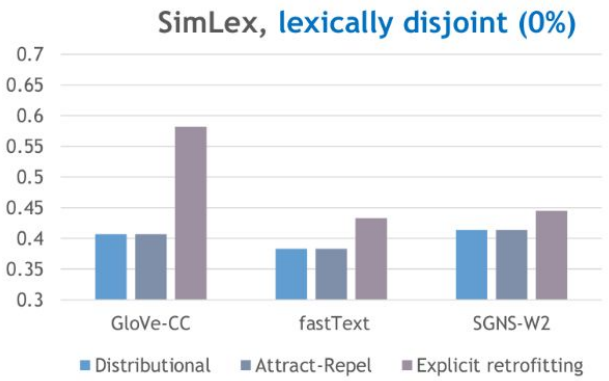
= 2

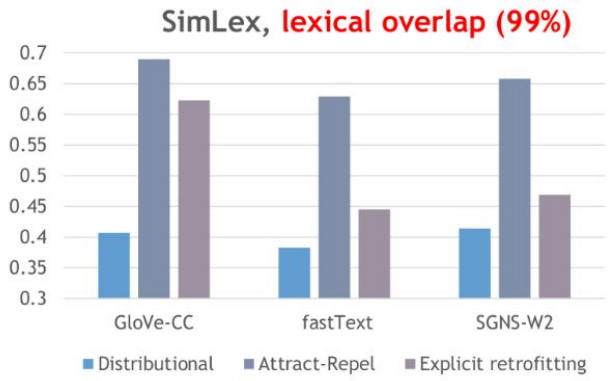
- **Regularization**

$$J_{REG} = \sum_{i=1}^{N} g(\mathbf{x}_1^i, f(\mathbf{x}_1^i)) + g(\mathbf{x}_2^i, f(\mathbf{x}_2^i))$$

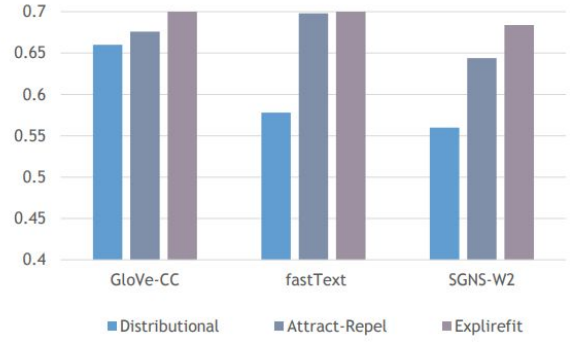
Explicit Retrofitting [Glavaš and Vulić, ACL-18]

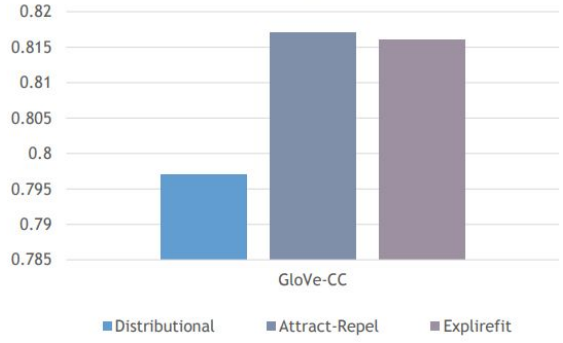
Intrinsic evaluation on word similarity in two insightful settings...





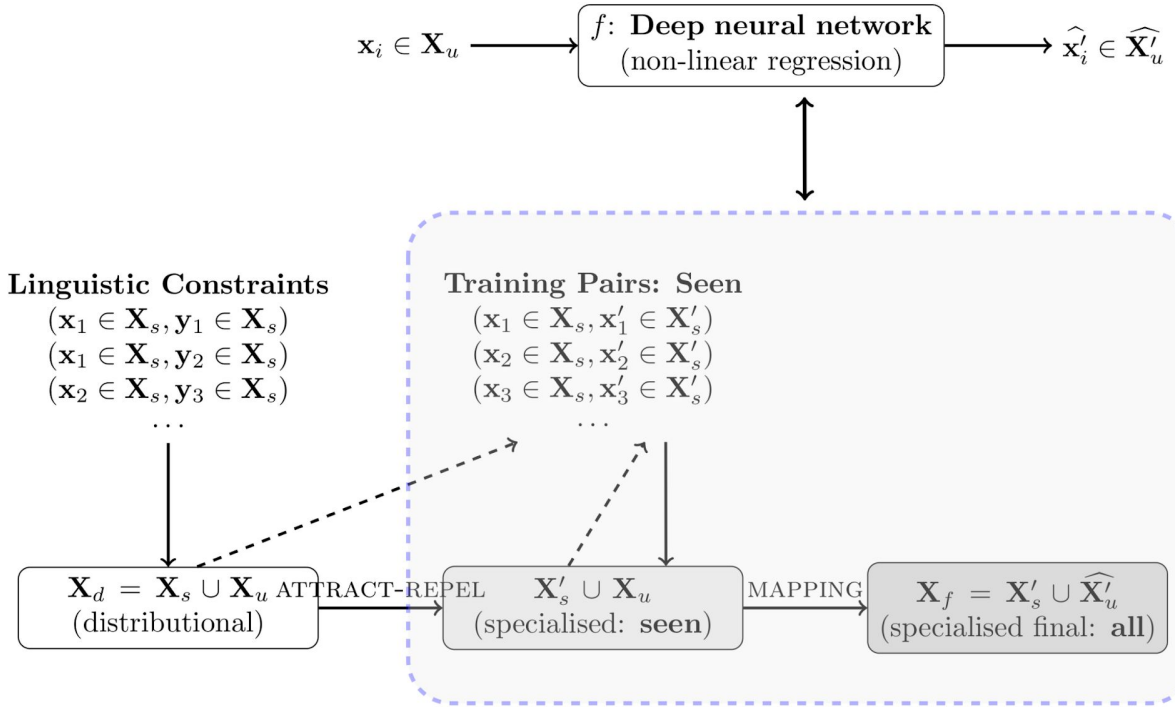
… and two suitable downstream tasks: 1. Lexical Simplification; 2. Dialogue State Tracking





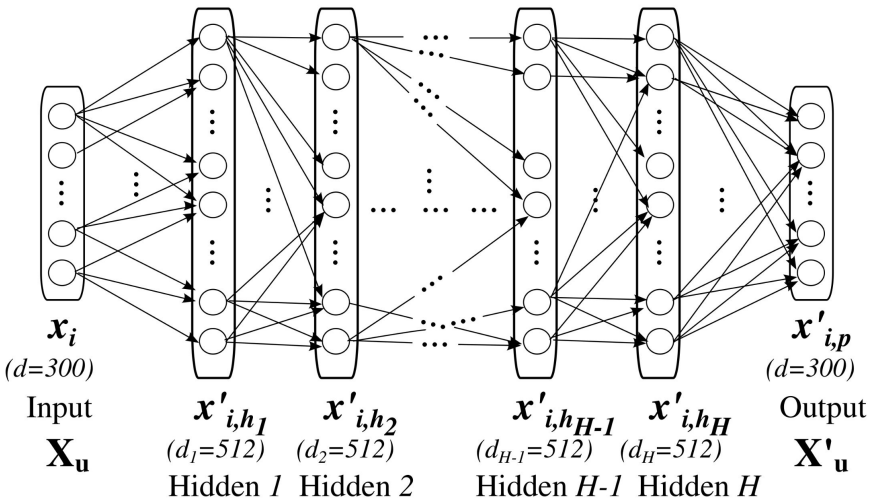
Post-Specialization [Vulić et al., NAACL-18]

The same goal as explicit retrofitting: full vocabulary specialisation



Post-Specialization [Vulić et al., NAACL-18]

Implemented as a deep feed-forward network (DFFN)



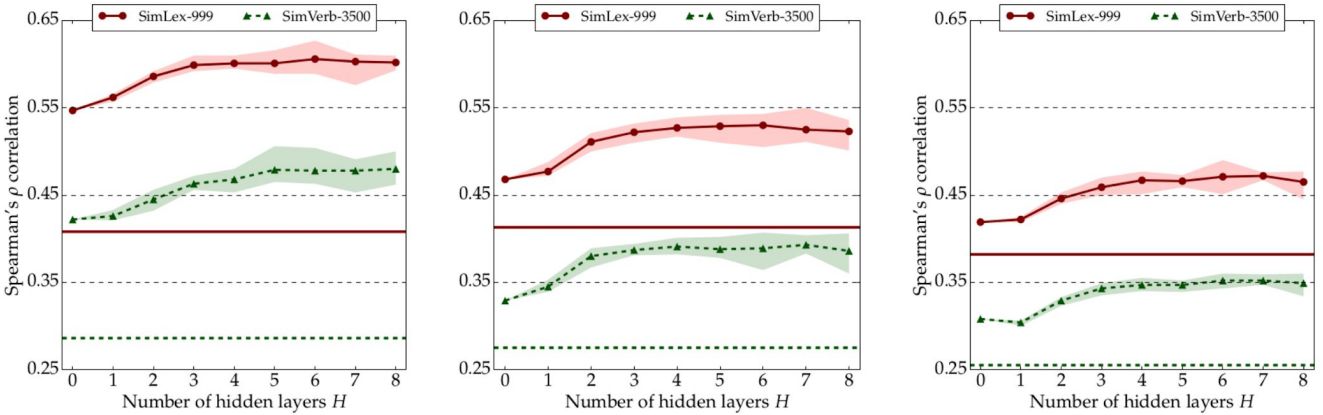
Loss functions and non-linearity matter

$$J_{\text{mse}} = \arg\min ||f(\mathbf{X}_s) - \mathbf{X}'_s||_F^2$$

$$J_{\text{MM}} = \sum_{i=1}^{N} \sum_{j \neq i}^{k} \tau\left(\delta_{mm} - cos\left(\widehat{\mathbf{x}'_i}, \mathbf{x}'_i\right) + cos\left(\widehat{\mathbf{x}'_i}, \mathbf{x}'_j\right)\right)$$

Post-Specialization: Some Results

	Setup: *hold-out*						Setup: *all*					
	GLOVE		SGNS-BOW2		FASTTEXT		GLOVE		SGNS-BOW2		FASTTEXT	
	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV
Distributional: $\mathbf{X}_d$	.408	.286	.414	.275	.383	.255	.408	.286	.414	.275	.383	.255
+AR specialisation: $\mathbf{X}'_s$	.408	.286	.414	.275	.383	.255	.690	.578	.658	.544	.629	.502
++Mapping unseen: $\mathbf{X}_f$												
LINEAR-MSE	.504	.384	.447	.309	.405	.285	.690	.578	.656	.551	.628	.502
NONLINEAR-MSE	.549	.407	.484	.344	.459	.329	.694	.586	.663	.556	.631	.506
LINEAR-MM	.548	.422	.468	.329	.419	.308	.697	.582	.663	.554	.628	.487
NONLINEAR-MM	**.603**	**.480**	**.531**	**.391**	**.471**	**.349**	**.705**	**.600**	**.667**	**.562**	**.638**	**.507**



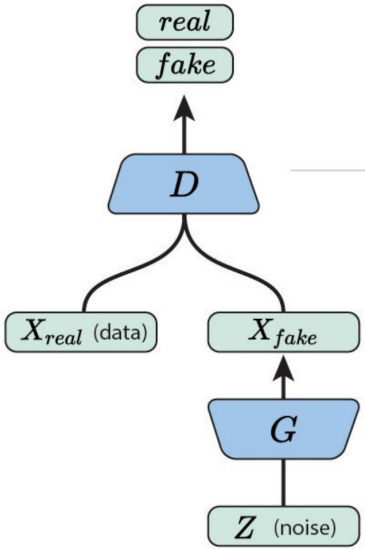
(a) GLOVE (b) SGNS-BOW2 (c) FASTTEXT

Adversarial Post-Specialization [Ponti et al., EMNLP-18]

Post-specialization with DFFN: **proof-of-concept**

Next-level: a more sophisticated model based on **adversarial training**

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.

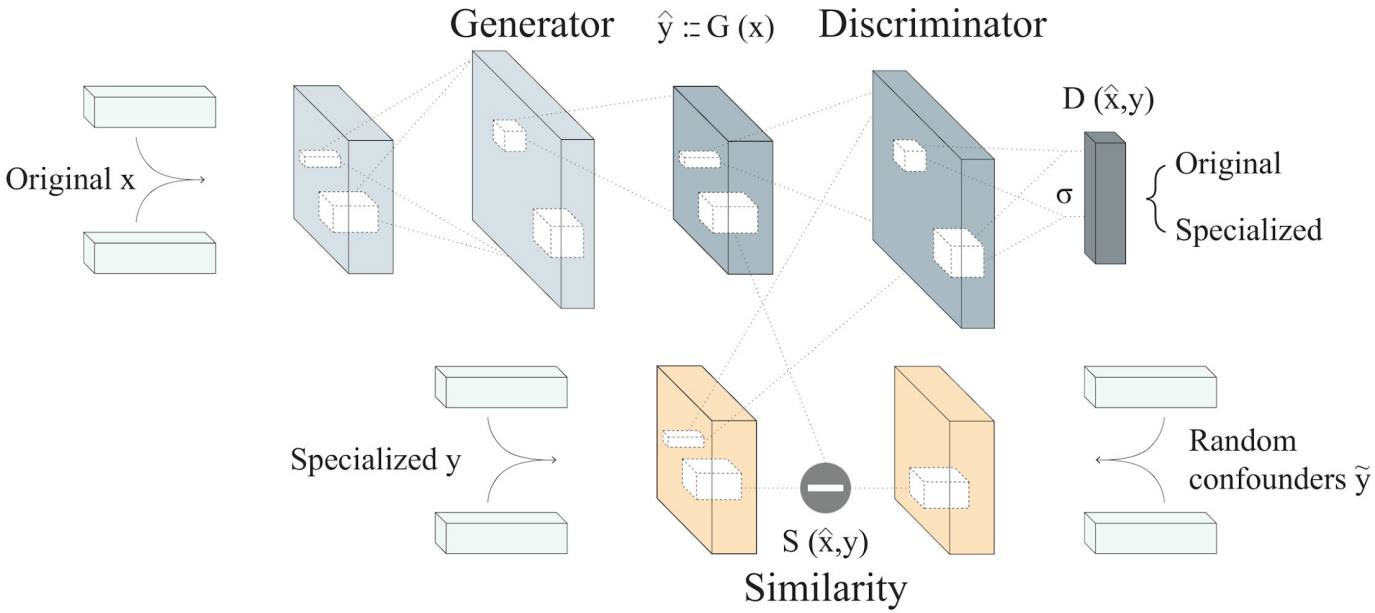


The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.

Adversarial Post-Specialization [Ponti et al., EMNLP-18]

A more sophisticated model based on **adversarial training**



Adversarial Post-Specialization [Ponti et al., EMNLP-18]

Tying three losses together:

$$\mathcal{L}_{MM} = \sum_{i=1}^{||\mathcal{V}_s||} \sum_{j \neq i}^{k} \tau[\delta_{MM} - \cos(G(\mathbf{x}_i^{(s)}; \theta_G), \mathbf{y}_i^{(s)}) + \cos(G(\mathbf{x}_i^{(s)}; \theta_G), \mathbf{y}_j^{(s)})]$$

$$\mathcal{L}_D = -\sum_{i=1}^{n} \log P(\text{spec} = 0 | G(\mathbf{x}_i; \theta_G); \theta_D) - \sum_{j=1}^{m} \log P(\text{spec} = 1 | \mathbf{y}_j; \theta_D)$$

$$\mathcal{L}_G = -\sum_{i=1}^{n} \log P(\text{spec} = 1 | G(\mathbf{x}_i; \theta_G); \theta_D) - \sum_{j=1}^{m} \log P(\text{spec} = 0 | \mathbf{y}_i; \theta_D)$$

Adversarial Post-Specialization [Ponti et al., EMNLP-18]

	Setting: DISJOINT					
	GLOVE-CC		FASTTEXT		SGNS-W2	
	SL	SV	SL	SV	SL	SV
Distributional (X)	.407	.280	.383	.247	.414	.272
Specialized: ATTRACT-REPEL	.407	.280	.383	.247	.414	.272
Post-Specialized: POST-DFFN	.645	.531	.503	.340	.553	.430
Post-Specialized: AUXGAN	**.652**	**.552**	**.513**	**.394**	**.581**	**.434**

Word Similarity Evaluation

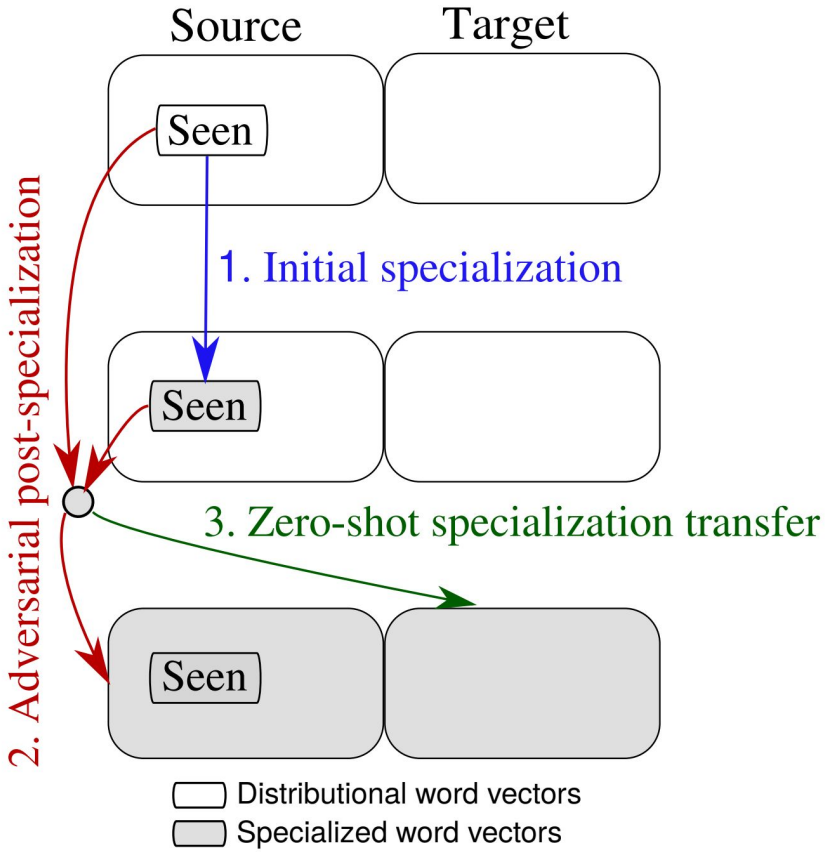
Lexical Simplification

	GLOVE-CC	FASTTEXT	SGNS-W2
Vector space	Acc	Acc	Acc
Distributional	.660	.578	.560
Specialized: AR	.676	.698	.644
Post-Specialized:			
POST-DFFN	**.723**	.723	.709
AUXGAN	.717	**.739**	**.721**

Dialogue State Tracking

GLOVE-CC word vectors	JGA
Distributional	.797
Specialized: ATTRACT-REPEL	.817
Post-Specialized: POST-DFFN	.829
Post-Specialized: AUXGAN	**.836**

Adversarial Post-Specialization: (Visual) Recap



Specialization for Asymmetric Relations

We say asymmetric relations, but in 99% of the cases it’s gonna be hypernymy… :)

Asymmetric Lexico-Semantic Relations

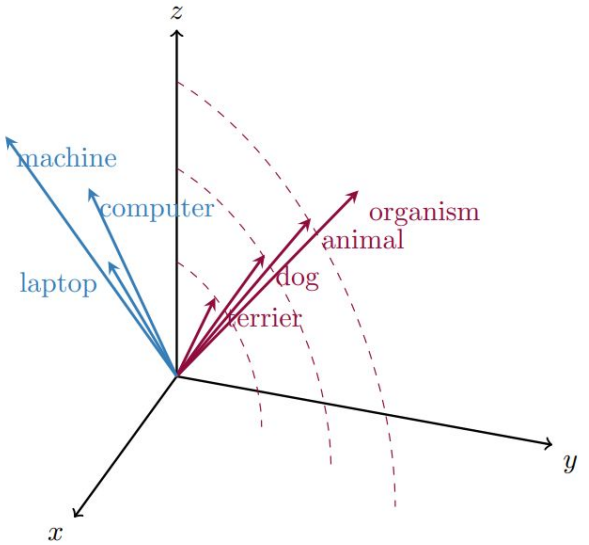
Semantic similarity, considered thus far, is a **symmetric relation**

Prominent **asymmetric** lexico-semantic relations

- Lexical entailment, aka (hyponymy-)hypernymy, “is-a” or “type-of” relation
    - 99% of work on asymmetric specialization focuses on LE
    - Applications: inference/entailment, taxonomy inductic
- Meronymy, aka “part-of” relation

Asymmetric relations are **hierarchical** in nature

- Symmetric relations: notion of **similarity**
- Asymmetric relations: notion of similarity + notion of **hierarchy**



Asymmetric Specialization: Model Typology

Joint specialization models

Retrofitting (post-processing) models

Post-specialisation models

Direct or **explicit specialization** models

+ Brief mention (related, but **not** strictly specialization models)
  + **Detection** of asymmetric relations from **distributional vectors**
  + Induction of hierarchies from scratch: **hyperbolic embeddings**

Quick Detour:
Detection of Asymmetric Relations

(Not really **specialization** but closely related, presented for completeness...)

Detection of asymmetric relations

Goal: predict the (LE) relation directly from distributional vectors
No change to the vectors, **no specialized vectors** as a result

Unsupervised: no parameters, just an asymmetric measure over dist. vectors
Distributional inclusion hypothesis
[Geffet & Dagan, ACL '05; Lenci & Benotto, *SEM '12]

Distributional informativeness / generality hypothesis
[Weeds et al., COLING '04; Santus et al., EACL '14]

Supervised: param. discriminative models, constraints = training examples
[Baroni et al., EACL 12; Shwartz et al., ACL 16; Glavaš & Ponzetto, EMNLP '17]

Unsupervised LE detection

Distributional generality/informativeness hypothesis: being more general, hypernym terms will appear in contexts of larger entropy than hyponyms

$$H(c) = -\sum_{i=1}^{n} p(f_i|c) \cdot log_2(p(f_i|c)) \Rightarrow E_{w_i} = Me_{j=1}^{N}\left(H_n(c_j)\right) \Rightarrow SLQS(w_1, w_2) = 1 - \frac{E_{w_1}}{E_{w_2}}$$

[Santus et al., EACL '14]

Distributional inclusion hypothesis: if word *v* entails word *w*, then *w* is expected to appear in all contexts in which *v* appears, but not vice-versa

$$\frac{\sum_{f \in F_u \cap F_v} min(w_u(f), w_v(f))}{\sum_{f \in F_u} w_u(f)}$$

[Lenci & Benotto, '12]

Overview and comparison of unsupervised LE detection:

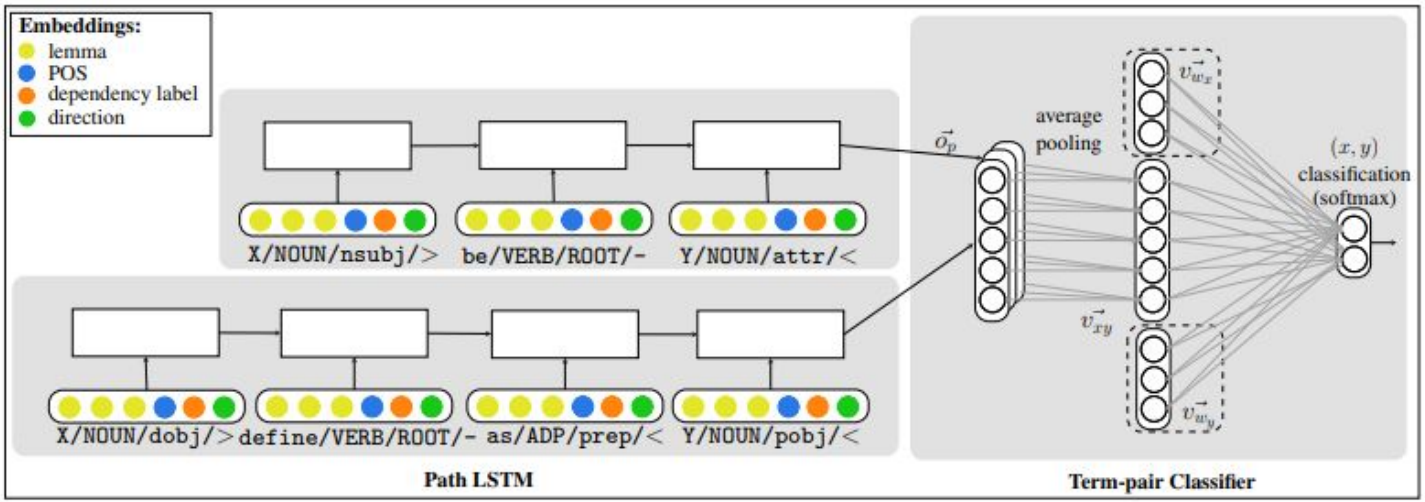
Shwartz, V., Santus, E., & Schlechtweg, D. (2017). Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. EACL '17 (pp. 65-75).

Supervised LE detection

HypeNet [Shwartz et al., ACL '16]

Encoding all corpus contexts where the term pairs appears

Classifying the encoding of all co-occurrence contexts



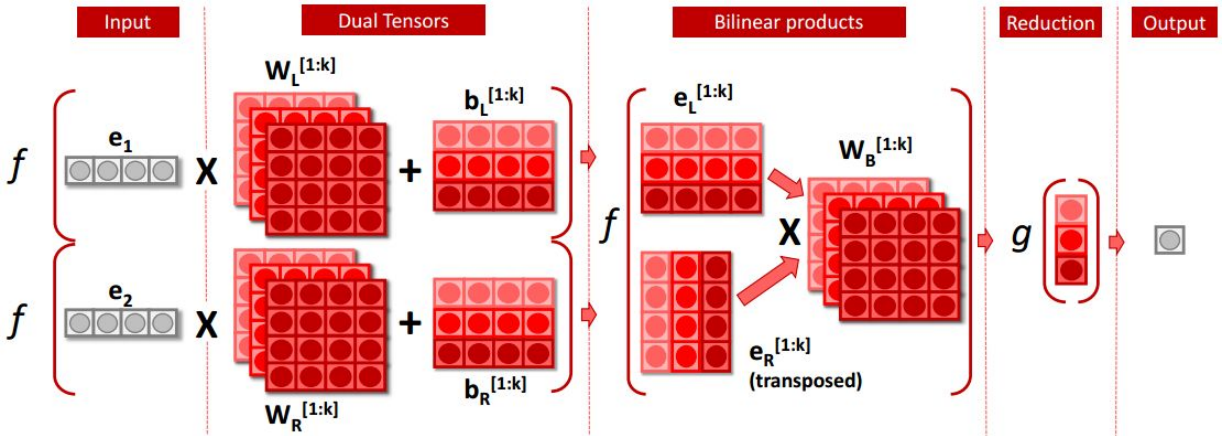
Supervised LE detection

Dual Tensor Model [Glavaš & Ponzetto, EMNLP '17]

Projecting (specializing?) distributional vectors differently depending on the position in the pair

Classifying the pair of projected vectors (LE; *meronymy*)

Related to *direct specialization*, but **no** specialized vectors produced



Quick Detour: Hyperbolic Embeddings

(Not really **specialization** but closely related, presented for completeness...)

Hyperbolic Embeddings

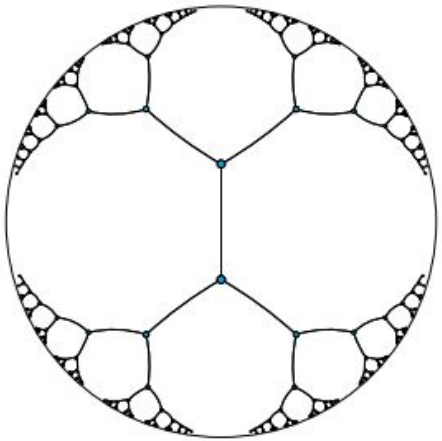
Assumption: Euclidean space not optimal for modelling hierarchies (and all our distributional models produce Euclidean spaces)

Hyperbolic geometry (non-Euclidean geometry, spaces of constant negative curvature) more suitable for modelling hierarchical data

- n-dimensional Poincaré ball
  [Nickel & Kiela, NeurIPS '17]
- Lorentz model of hyperbolic geometry
  [Nickel & Kiela, ICML '18]

$$d(\boldsymbol{u}, \boldsymbol{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{(1 - \|\boldsymbol{u}\|^2)(1 - \|\boldsymbol{v}\|^2)}\right)$$

Distances between points in the Poincare ball



2-dim. Poincaré ball
[Nickel & Kiela, NeurIPS '17]

Hyperbolic Embeddings

Input is a symbolic representation, a graph (i.e., a set of edges)

- Effectively a **graph-embedding model**, not vector specialization model

Optimization:

To compute Poincaré embeddings for a set of symbols $\mathcal{S} = \{x_i\}_{i=1}^n$, we are then interested in finding embeddings $\Theta = \{\boldsymbol{\theta}_i\}_{i=1}^n$, where $\boldsymbol{\theta}_i \in \mathcal{B}^d$. We assume we are given a problem-specific loss function $\mathcal{L}(\Theta)$ which encourages semantically similar objects to be close in the embedding space according to their Poincaré distance. To estimate Θ, we then solve the optimization problem

$$\Theta' \leftarrow \arg\min_{\Theta} \mathcal{L}(\Theta) \qquad \text{s.t. } \forall\, \boldsymbol{\theta}_i \in \Theta : \|\boldsymbol{\theta}_i\| < 1. \tag{2}$$

Concrete loss for taxonomy induction from WordNet (set of LE relations)

$$\mathcal{L}(\Theta) = \sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-d(\boldsymbol{u},\boldsymbol{v})}}{\sum_{\boldsymbol{v}'\in\mathcal{N}(u)} e^{-d(\boldsymbol{u},\boldsymbol{v}')}}$$

Solving via Riemannean optimization algorithms, e.g., RSGD

Asymmetric Specialization: Model Typology

Joint specialization models

Retrofitting (post-processing) models

Post-specialisation models

Direct or **explicit specialization** models

post-processing

+ Brief mention (related, but **not** strictly specialization models)
  + **Detection** of asymmetric relations from **distributional vectors**
  + Induction of hierarchies from scratch: **hyperbolic embeddings**

Joint versus Post-Processing

1. Joint "Heavy-Weight" Approaches

Induce representations *jointly*, by learning from contextual information while taking linguistic constraints into account.

2. Post-Processing Approaches

Inject semantic constraints into existing distributional vector spaces, treating them as black boxes.

The differences between two paradigms same as in specialization for similarity/symmetric relations

But now **model objectives** must add **asymmetric terms**

Joint LE Specialization

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 1: Dynamic Distance Margin Model
[Yu et al., IJCAI '15]

Probase **hypernymy**, **co-hypernymy**, and **co-hyponymy** constraints [Wu et al., SIGMOD '12] extracted via (Hearst and other) patterns from WaC

Triples: *x* = (*u*, *v*, *q*)
q = num. occurrences of (u, v) in corpus patterns
Two embeddings for each term *u*:
Hyponym embedding *O(u)*
Hypernym embedding *E(u)*

Strictly speaking not a joint model as there is no classic distributional co-occurrence objective,

Only the objective based on hypernymy and co-hyponymy constr.

$$f(x) = ||O(u) - E(v)||_1$$

$$J = \sum_{x=(u,v,q)} \sum_{j=1}^{q} max(0, f(x) - f(x'_j) + m(x, x'_j))$$

$$m(x, x') = log(q+1) - log(q'+1) = log\frac{q+1}{q'+1}$$

Joint LE Specialization

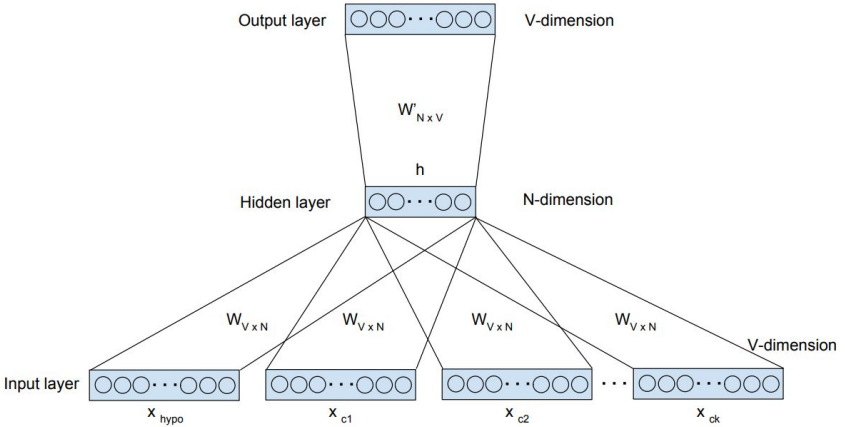
Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 2: Dynamic weighting network for predicting hypernym given hyponym and context
[Tuan et al., EMNLP-16]

Basically **CBOW with modified context definition**

$$h = W^\top \cdot \frac{1}{2k}(k \times x_{hypo} + x_{c_1} + x_{c_2} + ... + x_{c_k})$$
$$= \frac{1}{2k}(k \times v_{hypo} + v_{c_1} + v_{c_2} + ... + v_{c_k})$$

$$p(hype|hypo, c_1, c_2, .., c_k)$$
$$= \frac{e^{u_{hype}}}{\sum_{i=1}^{V} e^{u_i}}$$
$$= \frac{e^{v'^\top_{hype} \cdot \frac{1}{2k}(k \times v_{hypo} + \sum_{j=1}^{k} v_{c_j})}}{\sum_{i=1}^{V} e^{v'^\top_i \cdot \frac{1}{2k}(k \times v_{hypo} + \sum_{j=1}^{k} v_{c_j})}}$$



Joint LE Specialization: Directionality

Dynamic Distance Margin Model [Yu et al., IJCAI-15] and the Dynamic Weighting Network [Tuang et al., EMNLP-16]:

- Produce LE-informed embeddings, which, as features for classifiers (e.g., SVM) better predict LE relation

Major shortcoming: only predict the existence of LE for a pair of terms but cannot predict the directionality (i.e., which term is the *hypernym*)

This is because they lack **asymmetric terms** in their objectives

Joint LE Specialization

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 3: HyperVec Model
[Nguyen et al., EMNLP -17]

It augments (1) the distributional SkipGram objective with (2) LE-based objectives

First additional LE-based objective:

Bring hyponym vector closer to the hypernym vector when found to co-occur with the context word which is significantly more similar to the hyponym (by some margin)

Hyponym *w (e.g., bird)*
Hypernym *u (e.g., animal)*
Context word *c (e.g., wing)*

$$\mathbb{H}^+(w,c) = \{u \in \mathbb{W}(c) \cap \mathbb{H}(w) : cos(\vec{w},\vec{c}) - cos(\vec{u},\vec{c}) \geq \theta\}$$

$$\mathrm{L}_{(w,c)} = \frac{1}{\#(w,u)} \sum_{u \in \mathbb{H}^+(w,c)} \partial(\vec{w},\vec{u})$$

$L_{(w,c)}$: additional objective that forces the vector of the hyponym (*bird*) to get closer to the vector of hypernym *animal* (i.e., *bird* is an *animal*, but *animal* is not a *bird* in terms of the *wing* context)

Joint LE Specialization

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 3: HyperVec Model
[Nguyen et al., EMNLP -17]

It augments (1) the distributional SkipGram objective with (2) a set of LE constraints

Second additional LE-based objective:

Bring the hypernym vector closer to the hyponym vector when found to co-occur with the context word with which both are comparably similar (within some margin)

Hyponym *w (e.g., bird)*
Hypernym *u (e.g., animal)*
Context word *c (e.g., rights)*

$$\mathbb{H}^{-}(w,c) = \{v \in \mathbb{W}(c) \cap \mathbb{H}(w) : cos(\vec{w},\vec{c}) - cos(\vec{v},\vec{c}) < \theta\}$$

$$\mathrm{L}_{(v,w,c)} = \sum\nolimits_{v \in \mathbb{H}^{-}(w,c)} \partial(\vec{v},\vec{w})$$

$L_{(v, w,c)}$: additional objective that forces the vector of the hypernym (*animal*) to get closer to the vector of the hyponym *bird* in the context of *rights* (i.e., *animal rights* are *bird rights*)

Joint LE Specialization

Joint = modifying the distributional objective directly
Joint = distributional-based objective + resource-based objective

Example 3: HyperVec Model
[Nguyen et al., EMNLP -17]

It augments (1) the distributional SkipGram objective with (2) a set of LE constraints

Overall objective:

Skip-Gram negative sampling objective (SGNS) $J_{(w,c)}$ augmented with $L_{(w,c)}$ and $L_{(v, w,c)}$

$$\mathrm{J}_{(w,v,c)} = \mathrm{J}_{(w,c)} + \mathrm{L}_{(w,c)} + \mathrm{L}_{(v,w,c)}$$

$$\mathrm{J} = \sum_{w \in V_W} \sum_{c \in V_C} \mathrm{J}_{(w,v,c)}$$

Joint LE Specialization Models: Problems

They are “**heavy-weight”** and **demanding:**

- Training from large text corpora from scratch any time we want to change something
- Tied to the underlying distributional architecture and objectives (e.g., CBOW or SGNS)

Long training times and **less-competitive performance**:

- Effective balancing between the two sources of information is non-trivial
  - In case of LE, there are additional components capturing asymmetry
- Extortionate computational complexity

Can we apply asymmetric (i.e., LE) specialization as a post-processing step as well?
Retrofitting and pals...

Lexical Entailment Attract-Repel (LEAR) [Vulić & Mrkšić, NAACL-18]

LEAR is a **retrofitting model** that post-hoc specializers arbitrary distributional vectors for lexical entailment. It is an LE-based extension of AR

AR: *Attract* objective from synonymy constraints

Repel objective from antonymy constraints

Regularization objective for all constraints

LEAR:

+ Attract objective from LE constraints
+ Asymmetric norm-based objective from LE constraints

Lexical Entailment Attract-Repel (LEAR) [Vulić & Mrkšić, NAACL-18]

LEAR is a **retrofitting model** that post-hoc specializers arbitrary distributional vectors for lexical entailment. It is an LE-based extension of AR

Same Attract & Repel objectives as before…

$$\sum_{k=1}^{K} [\, \tau \left(\delta_{att} + \cos \left(\mathbf{x}_l^{(k)}, \mathbf{t}_l^{(k)} \right) - \cos \left(\mathbf{x}_l^{(k)}, \mathbf{x}_r^{(k)} \right) \right) + \tau \left(\delta_{att} + \cos \left(\mathbf{x}_r^{(k)}, \mathbf{t}_r^{(k)} \right) - \cos \left(\mathbf{x}_l^{(k)}, \mathbf{x}_r^{(k)} \right) \right)].$$

(Attract objective, Repel objective is analogous)

New **asymmetric** objective

$$LE(\mathcal{B}_{Le}) = \sum_{k=1}^{K} \frac{\|\mathbf{x}_l^{(k)}\| - \|\mathbf{x}_r^{(k)}\|}{\|\mathbf{x}_l^{(k)}\| + \|\mathbf{x}_r^{(k)}\|}$$

Overall objective

$$\begin{aligned} C(\mathcal{B}_A, T_A, \mathcal{B}_R, T_R, \mathcal{B}_L, T_L) \;=\; & Att(\mathcal{B}_S, T_S) + \ldots \\ & + \; Rep(\mathcal{B}_A, T_A) \; + \; Reg(\mathcal{B}_A, \mathcal{B}_R, \mathcal{B}_L) \; + \; \ldots \\ & + \; Att(\mathcal{B}_L, T_L) \; + \; LE_j(B_L) \end{aligned}$$

Lexical Entailment Retrofitting: Shortcoming

LEAR is a **retrofitting model** that post-hoc specializers arbitrary distributional vectors for lexical entailment. It is an LE-based extension of AR

As a retrofitting model, LEAR changes/specializes **only** the vectors of words found in the constraints (synonyms, antonyms, LE pairs)

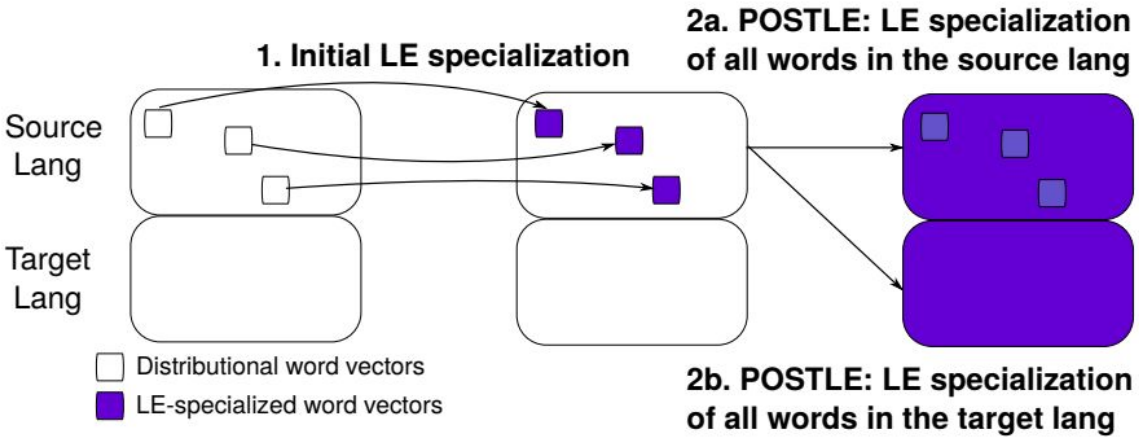
We would like to do the **full LE specialization**, i.e., specialize the whole distributional space for lexical entailment

1) **Post-specialization** for lexical entailment
   [Kamath et al., RepL4NLP-19]

2) **Direct/explicit specialization** for lexical entailment
   [Glavaš & Vulić, ACL-19]

Post-specialization for Lexical Entailment [Kamath et al., RepL4NLP-19]

POSTLE is a **post-specialization model for lexical entailment** which captures a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.

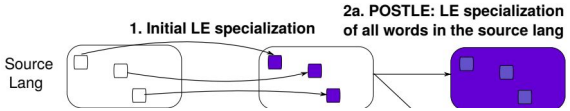


Mapping: feed-forward network (simple and GAN-augmented architectures)

Training examples: pairs (**x**, **x'**) where **x** is the distributional and **x'** is LEAR-retrofitted vector of some constraint word

Post-specialization for Lexical Entailment [Kamath et al., RepL4NLP-19]

POSTLE is a **post-specialization model for lexical entailment** which captures a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.



Mapping architecture: adversarial architecture

Generator: Feed forward network $G(\mathbf{x}_s; \theta_G)$

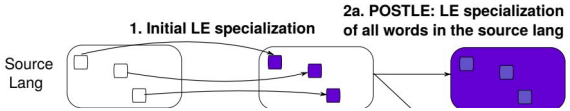
- Given distributional vector $\mathbf{x}_s$, generate/predict LEAR-retrofitted vector $\mathbf{y}_s$

- Mapping loss compares both direction and norms of $G(\mathbf{x}_s; \theta_G)$ and $\mathbf{y}_s$

$$\mathcal{L}_S = dcos\left(G(\mathbf{x}_s; \theta_G), \mathbf{y}_s\right) + \delta_n \left| \|G(\mathbf{x}_s; \theta_G)\| - \|\mathbf{y}_s\| \right|$$

Post-specialization for Lexical Entailment [Kamath et al., RepL4NLP-19]

POSTLE is a **post-specialization model for lexical entailment** which captures a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.



Mapping architecture: adversarial architecture

Discriminator: Feed forward network $D(\mathbf{x}; \theta_D)$

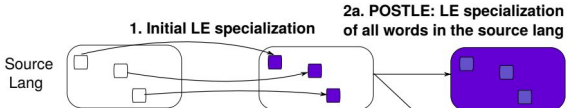
- Given a vector **x**, predict if it was generated by the $G(\mathbf{x}; \theta_G)$ or is a true LEAR-retrofitted vector $\mathbf{y}_s$

- Loss: binary misclassification loss (negative log-likelihood)

$$\mathcal{L}_D = -\sum_{s=1}^{N} \log P(\text{spec} = 0 | G(\mathbf{x}_s; \theta_G); \theta_D)$$
$$-\sum_{s=1}^{M} \log P(\text{spec} = 1 | \mathbf{y}_s; \theta_D)$$

Post-specialization for Lexical Entailment [Kamath et al., RepL4NLP-19]

POSTLE is a **post-specialization model for lexical entailment** which captures a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.



Mapping architecture: adversarial architecture

Generator: Feed forward network $G(\mathbf{x}_s; \theta_G)$

- Additionally needs to confuse the discriminator

- **Additional loss**: negative log likelihoods of discriminator's correct classifications

$$\mathcal{L}_G = -\sum_{s=1}^{N} \log P(\text{spec} = 1 | G(\mathbf{x}_s; \theta_G); \theta_D)$$

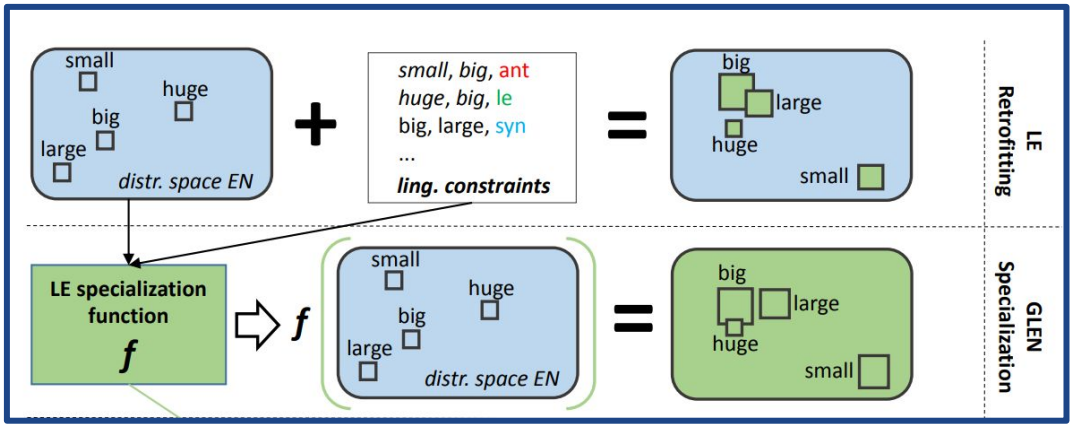
$$-\sum_{s=1}^{M} \log P(\text{spec} = 0 | \mathbf{y}_s; \theta_D)$$

Direct / Explicit Retrofitting for LE [Glavaš & Vulić, ACL-19]

Idea: can we avoid the retrofitting step altogether and learn an explicit LE specialization function directly from constraints?

Idea: use constraints directly as training examples

GLEN (Generalized Lexical)is a **post-specialization model for lexical entailment** which directly learns a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.



Direct / Explicit Retrofitting for LE [Glavaš & Vulić, ACL-19]

GLEN (Generalized Lexical ENtailment) is a **post-specialization model for lexical entailment** which directly learns a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.

What is the LE-specialization function ***f*** ?

As in POSTLE, it is a simple feed-forward net

$$h^i(\mathbf{x};\theta_i) = \psi\left(h^{i-1}(\mathbf{x},\theta_{i-1})\mathbf{W}^i + \mathbf{b}^i\right)$$

But unlike in POSTLE, it's trained directly with constraints (syn, ant, LE) as training examples

Objectives functions similar to those in LEAR and POSTLE

Direct / Explicit Retrofitting for LE [Glavaš & Vulić, ACL-19]

GLEN (Generalized Lexical ENtailment) is a **post-specialization model for lexical entailment** which directly learns a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.

Asymmetric norm-based loss (applied only for batches of LE constraints)

$$l_a = \sum_{k=1}^{K} \tau \left(\delta_a - d_N\left(f(\mathbf{x}_1^k), f(\mathbf{y}_1^k)\right) + d_N\left(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\right)\right) + \tau \left(\delta_a - d_N\left(f(\mathbf{y}_2^k), f(\mathbf{x}_2^k)\right) + d_N\left(f(\mathbf{x}_1^k), f(\mathbf{x}_2^k)\right)\right)$$

with d_N as the asymmetric norm-based distance of the vectors

$$d_N(\mathbf{x}_1, \mathbf{x}_2) = \frac{\|\mathbf{x}_1\| - \|\mathbf{x}_2\|}{\|\mathbf{x}_1\| + \|\mathbf{x}_2\|}$$

Direct / Explicit Retrofitting for LE [Glavaš & Vulić, ACL-19]

GLEN (Generalized Lexical ENtailment) is a **post-specialization model for lexical entailment** which directly learns a general/explicit LE-specialization function by using vectors of constraint words and their retrofitted vectors as training examples.

Symmetric direction-based “Similarity” (“Attract”) loss

Symmetric direction-based “Dissimilarity” (“Repel”) loss

Semantic regularization loss

Different combinations of losses for different types of constraints:

LE (hypo-hyper) $J_E = l_s(E) + \lambda_a \cdot l_a(E) + \lambda_r \cdot l_r(E);$

Synonyms $J_S = l_s(S) + \lambda_r \cdot l_r(S);$

Antonyms $J_A = l_d(A) + \lambda_r \cdot l_r(A),$

Evaluation for LE-specialized embeddings

LE detection

- Detect LE against other relations: synonymy, antonymy, co-hyponymy, ...

LE directionality

- LE (hyponym-hypernym) vs. inverse LE (hypernym-hyponym)

Graded LE

- The degree to which w_1 is a ***type of*** w_2

Evaluation: LE detection

BLESS dataset [Baroni & Lenci, GEMNLS-11]

- 26K word pairs
- LE vs. meronymy, coordination, event, attribute, random (unbalanced)
- BIBLESS [Kiela et al, ACL-15] : detection and directionality

EVALuation dataset [Santus et al., LDL-15]

- 13.5K word pairs
- LE vs. meronymy, attribute, synonymy, antonymy (unbalanced)

Weeds dataset [Weeds et al., COLING-14]

- LE vs. co-hyponymy, 3K pairs, balanced

Benotto dataset [Benotto, thesis-15]

- LE vs. synonyms and antonyms, 5K, ~balanced

Evaluation: LE detection

Previous datasets effectively only evaluation datasets

- Too small to be used for training neural models

HypeNet dataset [Shwartz et al., ACL-16]

- WordNet + other external DBs (e.g., Freebase)
- Binary: LE vs. all other relations
- 50K train, 3.5K validation, ~18K test

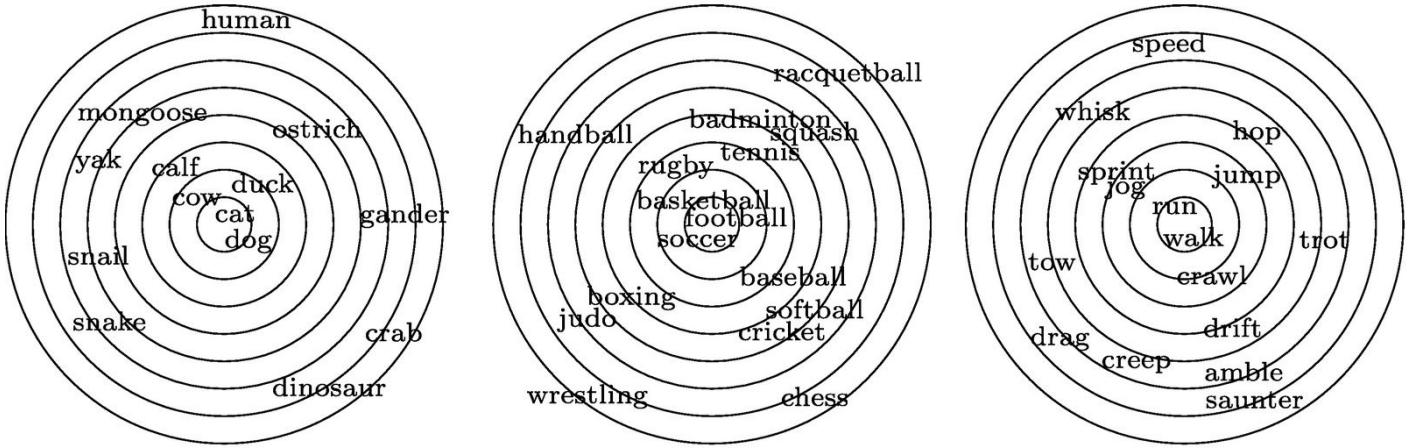
WN-Hy dataset [Glavaš & Ponzetto, EMNLP-17]

- WordNet, LE vs. synonymy, antonymy, meronyms
- 103K train, 15K validation, 30K test

Evaluation: Graded LE

Can LE be seen as a graded relation? Or is it strictly binary?

- “To which degree is A a type of B”?
- Is *“bread”* more of a *“food”* than *“cinnamon”*?



(a) ANIMAL (b) SPORT (c) TO MOVE

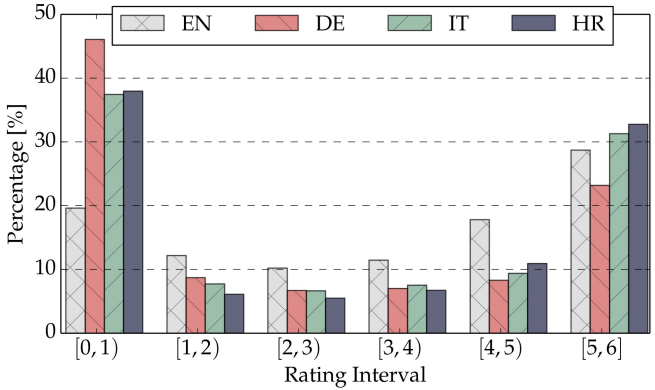
Image from [Vulić et al., CompLing-17]

Evaluation: Graded LE

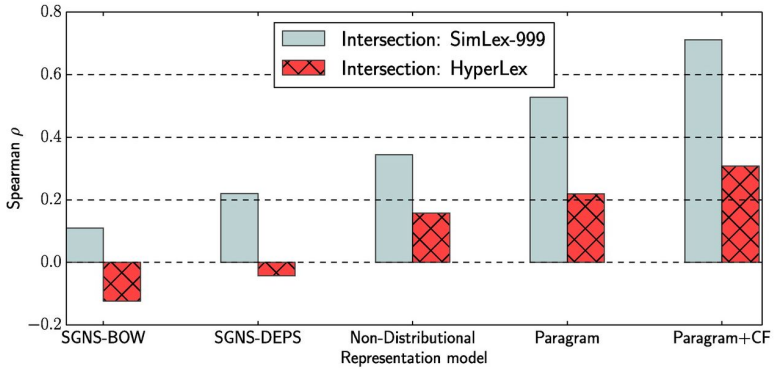
HyperLex [Vulić et al., CompLing-17]:

- 2,616 pairs (verb and noun pairs) annotated for graded LE (0-6 scale)
- Pairs in relations: LE-N, inverse LE-N, co-hyp, mero, syn, ant, no-rel (random)
- **De facto standard for evaluating LE-specialized word vectors**

- Annotations -> graded nature of LE



Graded LE ≠ sem. similarity



Evaluation for LE Specialization: Some Results

LEAR (retrofitting) [Vulić & Mrkšić, NAACL-18] vs. previous LE-scoring models on HyperLex

Caveat:

- LEAR LE-specializes only vectors of words seen in constraints
- 99% of HyperLex words can be found in LEAR constraints
- Favorable setting for LEAR (unlikely to hold in downstream tasks)

	All
FREQ-RATIO	0.279
SGNS (COS)	0.205
SLQS-SIM	0.228
VISUAL	0.209
WN-BEST	0.234
WORD2GAUSS	0.206
SIM-SPEC	0.320
ORDER-EMB	0.191
POINCARÉ (nouns)	0.512
HYPERVEC	0.540
Best LEAR	**0.686**

Evaluation for LE Specialization: Some Results

LEAR (retrofitting) [Vulić & Mrkšić, NAACL-18] vs. **GLEN** (explicit/direct specialization) [Glavaš & Vulić, ACL-19] on HyperLex

Setup	0%	10%	30%	50%	70%	90%	100%
LEAR	.174	.188	.273	.438	**.548**	**.634**	**.682**
GLEN	**.481**	**.485**	**.478**	**.474**	.506	.504	.520

Table 1: Spearman correlation for GLEN, compared with LEAR (Vulić and Mrkšić, 2018), on HyperLex, for different word coverage settings (i.e., percentages of Hyperlex words *seen* in constraints in training).

Evaluation for LE Specialization: Some Results

LEAR (retrofitting) [Vulić & Mrkšić, NAACL-18] vs. **POSTLE** (post-specialization, learning from retrofitted vectors) [Kamath et al., RepL4NLP-19] on HyperLex

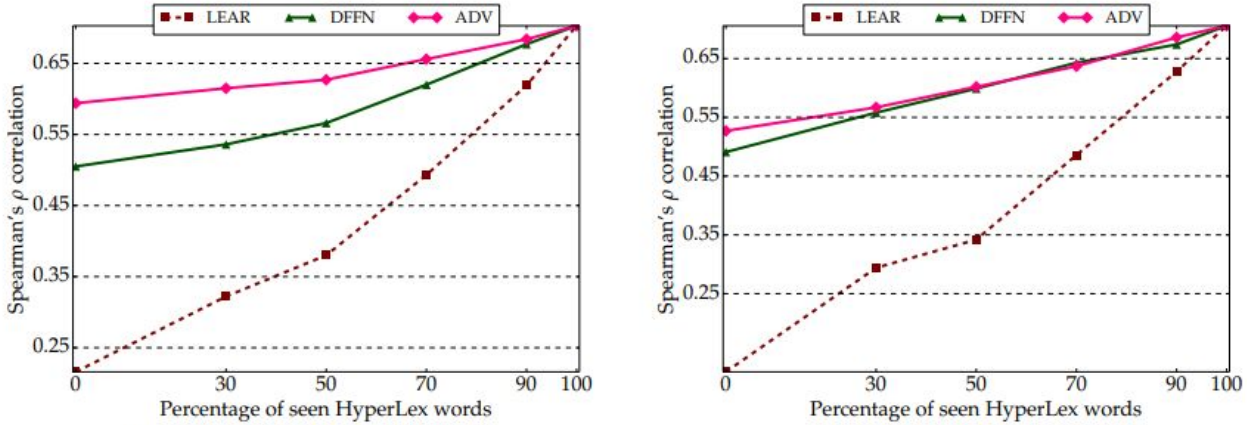


Figure 2: Spearman's ρ correlation scores for two input distributional spaces on the noun portion of HyperLex (2,163 concept pairs) conditioned on the number of test words covered (i.e., *seen*) in the external lexical resource. Similar patterns are observed on the full HyperLex dataset. Two other baseline models report the following scores on the noun portion of HyperLex in the 100% setting: 0.512 (Nickel and Kiela, 2017); 0.540 (Nguyen et al., 2017).

Specialization for Arbitrary Relations

Semantic Specialization is a General Framework

So far, we focused on standard lexico-semantic relations

- **Fine-tuning** word vectors for these relations expected to be beneficial for a wide(r) range of downstream tasks

But the presented frameworks are **general** and can be applied for any relation

- Need: relation-specific constraints
- Specific relations useful for a narrower set of downstream tasks

Some examples:

- Specialization for morphological relatedness [Vulić et al., ACL-17]
- Specialization for sentiment [Yu et al., EMNLP-17]
- Specialization for affect [Khosla et al. COLING-18]
- Debiasing word vectors via direct specialization [Lauscher et al., 19]

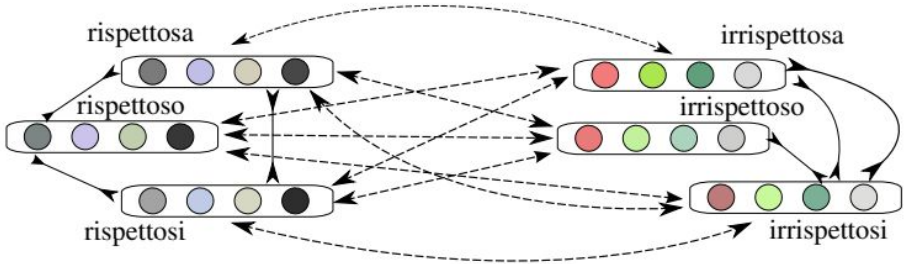
Morph-fitting [Vulić et al., ACL-17]

Morph-fitting: specializing the distributional space with constraints induced from easy-to-design **language-specific morphological rules**

- meaning("*acquire*") ~ meaning ("*acquires*")
- meaning("*expensive*") ≠ meaning("***in****expensive*")

Idea:

1. Language-specific **morph-rules** create "Attract" and "Repel" constraints
2. Apply the usual suspect -- **retrofitting with AR** -- to specialize the space



Morph-fitting [Vulić et al., ACL-17]

Morph-fitting: specializing the distributional space with constraints induced from easy-to-design **language-specific morphological rules**

Language-specific rules for creating constraints:

Attract constraints: inflectional morphology

EN: w2 = w1 + ing/ed/s: (look, looks), (look, looking), (look, looked)

IT, EN, RU: reg. plural formation: (libro, libri), (Buch, Bücher), (собака, собаки)

verb conjugation: (dare, diamo), (machen, macht), (работать, работаю)

grammatical gender: (bianco, bianca), (sein, seine), (синий, синее)

Repel constraints: derivational morphology

EN: w2 = {*dis*, *il*, *un*, *in*, *im*, *ir*, *mis*, *non*, *anti*} + w1

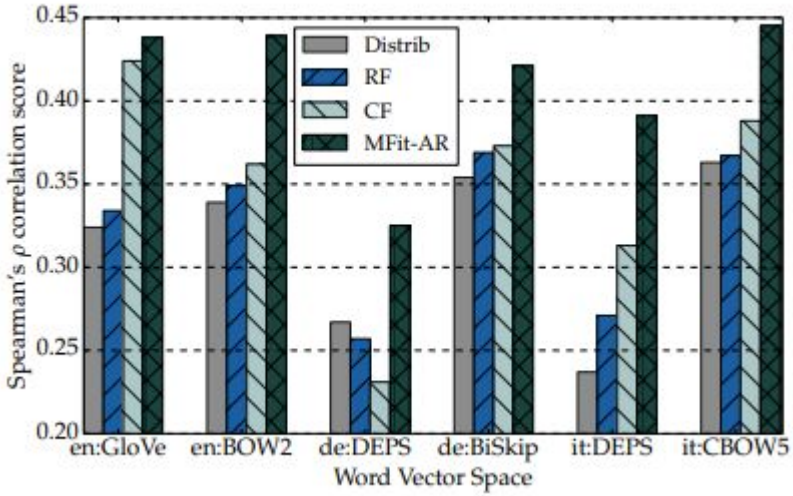
IT: w2 = {in, ir, im, anti} + w1

...

Morph-fitting [Vulić et al., ACL-17]

Morph-fitting: specializing the distributional space with constraints induced from easy-to-design **language-specific morphological rules**

Vectors	Evaluation	
	SimLex-999	SimVerb-3500
1. SG-BOW2-PW (300) (Mikolov et al., 2013)	.339 → **.439**	.277 → **.381**
2. GloVe-6B (300) (Pennington et al., 2014)	.324 → **.438**	.286 → **.405**
3. Count-SVD (500) (Baroni et al., 2014)	.267 → **.360**	.199 → **.301**
4. SG-DEPS-PW (300) (Levy and Goldberg, 2014)	.376 → **.434**	.313 → **.418**
5. SG-DEPS-8B (500) (Bansal et al., 2014)	.373 → **.441**	.356 → **.473**
6. MultiCCA-EN (512) (Faruqui and Dyer, 2014)	.314 → **.391**	.296 → **.354**
7. BiSkip-EN (256) (Luong et al., 2015)	.276 → **.356**	.260 → **.333**
8. SG-BOW2-8B (500) (Schwartz et al., 2015)	.373 → **.440**	.348 → **.441**
9. SymPat-Emb (500) (Schwartz et al., 2016)	.381 → **.442**	.284 → **.373**
10. Context2Vec (600) (Melamud et al., 2016)	.371 → **.440**	.388 → **.459**



Refining Word Embeddings for Sentiment [Yu et al., EMNLP-17]

Sentiment-based specialization constraints come from a sentiment lexicon:

- Valence scores from the E-ANEW lexicon [Warriner et al., BRM-13]

A type of a **retrofitting model** based on sentiment neighbourhoods

1. Retrieve *k* nearest neighbours according to the valence/sentiment score
   a. Rank them and assign weights according to where in the ranking it is
2. Minimize the weighted distance to each sentiment neighbour
3. Regularize: disallow vectors to deviate too much from original positions

$$\arg\min \Phi(V) =$$

$$\arg\min \sum_{i=1}^{n} \left[\alpha dist(v_i^{t+1}, v_i^t) + \beta \sum_{j=1}^{k} w_{ij} dist(v_i^{t+1}, v_j^t) \right]$$

Refining Word Embeddings for Sentiment [Yu et al., EMNLP-17]

A type of a **retrofitting model** based on sentiment neighbourhoods

Sentiment-retrofitting of vectors then bring gains in downstream sentiment classification, regardless of the classifier architecture and source vecs

CNN		
- Word2vec	48.0	87.2
- GloVe	46.4	85.7
- Re(Word2vec)	**48.8**	**87.9**
- Re(GloVe)	47.7	87.5
- HyRank	47.3	87.6

Bi-LSTM		
- Word2vec	48.8	86.3
- GloVe	49.1	87.5
- Re(Word2vec)	49.6	88.2
- Re(GloVe)	**49.7**	**88.6**
- HyRank	49.0	87.3

Tree-LSTM		
- Word2vec	48.8	86.7
- GloVe	51.8	89.1
- Re(Word2vec)	50.1	88.3
- Re(GloVe)	**54.0**	**90.3**
- HyRank	49.2	88.2

Aff2Vec [Khosla et al., COLING-18]

Aff2Vec: retrofitting word vectors based on affect-based constraints

Affect-based specialization based on E-ANEW lexicon [Warriner et al., BRM-13]

- 14K English words annotated with 3 scores: Valence, Arousal, Dominance

Affect-Append: concatenate 3-dim V-A-D vectors to distributional vectors of words

Affect-Strength: modify the Retrofitting [Faruqui et al., NAACL-15] algorithm so that the association weight is the *"affectual"* similarity between words

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

$$\beta_{ij} \rightarrow 1 - \frac{\|a_i - a_j\|}{\sqrt{\sum_{f=1}^{F} max\_dist_f^2}}$$

Aff2Vec [Khosla et al., COLING-18]

Affect-related downstream tasks:

- FFT (formality, frustration, politeness) detection
- Personality detection
- Emotion classification

Model	FFP-Prediction			Personality Detection					SA	EMO-INT			
	MSE ($X10^{-3}$)			Acc. (%)					Acc. (%)	Pearson's ρ ($X10^{-2}$)			
	FOR	FRU	POL	EXT	NEU	AGR	CON	OPEN	DAN	ANG	FEA	JOY	SAD
GloVe	27.59	32.40	21.89	**56.08**	55.25	56.06	**57.32**	59.14	83.1	70.98	71.19	65.85	73.30
⊕ Affect	27.72	28.76	22.02	51.47	57.41	56.09	55.06	**62.08**	84.3	70.91	71.72	66.26	**73.58**
+ Retrofitting	27.44	29.35	21.75	55.79	59.67	55.59	56.89	59.67	82.7	72.10	71.86	**67.11**	73.14
+ Retrofitting ⊕ Affect	28.33	**27.91**	22.24	55.01	56.43	**57.48**	53.04	61.12	83.7	**72.38**	**72.53**	66.29	72.76
+ Counterfitting	**25.66**	29.20	22.90	55.11	58.32	55.41	53.89	60.36	84.2	70.45	68.95	65.27	72.63
+ Counterfitting ⊕ Affect	28.89	32.46	**21.64**	52.12	**60.03**	56.53	54.93	59.51	**84.4**	70.20	70.43	65.81	72.37

DebiasNet [Lauscher et al., 19]

DebiasNet is an direct/explicit specialization model for removing arbitrarily defined biases from distributional word vectors

Given a **bias specification** (two target lists and attributes w.r.t. the bias exists), fine-tune the distributional vectors so that the bias no longer holds

Example bias specifications

- gender bias w.r.t. science vs. art;

Initial T_1	*science*, *technology*, *physics*, *chemistry*, *Einstein*, *NASA*, *experiment*, *astronomy*
Initial T_2	*poetry*, *art*, *Shakespeare*, *dance*, *literature*, *novel*, *symphony*, *drama*
Initial A_1	*brother*, *father*, *uncle*, *grandfather*, *son*, *he*, *his*, *him*
Initial A_2	*sister*, *mother*, *aunt*, *grandmother*, *daughter*, *she*, *hers*, *her*

DebiasNet [Lauscher et al., 19]

DebiasNet is an direct/explicit specialization model for removing arbitrarily defined biases from distributional word vectors

Given a **bias specification** (two target lists and attributes w.r.t. the bias exists), fine-tune the distributional vectors so that the bias no longer holds

Example bias specifications

- sentiment bias w.r.t. insects vs. flowers

Initial	T_1	*aster clover hyacinth marigold poppy azalea crocus iris orchid rose blue-bell daffodil lilac pansy tulip buttercup daisy lily peony violet carnation gladiola magnolia petunia zinnia*
	T_2	*ant caterpillar flea locust spider bedbug centipede fly maggot tarantula bee cockroach gnat mosquito termite beetle cricket hornet moth wasp blackfly dragonfly horsefly roach weevil*
	A_1	*caress freedom health love peace cheer friend heaven loyal pleasure diamond gentle honest lucky rainbow diploma gift honor miracle sunrise family happy laughter paradise vacation*
	A_2	*abuse crash filth murder sickness accident death grief poison stink assault disaster hatred pollute tragedy divorce jail poverty ugly cancer kill rotten vomit agony prison*

DebiasNet [Lauscher et al., 19]

DebiasNet is an direct/explicit specialization model for removing arbitrarily defined biases from distributional word vectors

1. Rely on semantic specialization of word vectors for similarity to augment bias spec.
   - Adversarial similarity post-specialization model [Ponti et al., EMNLP-18]

Initial T_1	*science, technology, physics, chemistry, Einstein, NASA, experiment, astronomy*
Initial T_2	*poetry, art, Shakespeare, dance, literature, novel, symphony, drama*
Initial A_1	*brother, father, uncle, grandfather, son, he, his, him*
Initial A_2	*sister, mother, aunt, grandmother, daughter, she, hers, her*

Augmentation T_1	*automation, radiochemistry, test, biophysics, learning, electrodynamics, biochemistry, astrophysics, astrometry*
Augmentation T_1	*orchestra, artistry, dramaturgy, poesy, philharmonic, craft, untried, hop, poem, dancing, dissertation, treatise*
Augmentation A_1	*beget, buddy, forefather, man, nephew, own, himself, theirs, boy, crony, cousin, grandpa, granddad*
Augmentation A_2	*niece, girl, parent, grandma, granny, woman, theirs, sire, auntie, sibling, herself, jealously, stepmother, wife*

DebiasNet [Lauscher et al., 19]

DebiasNet is an direct/explicit specialization model for removing arbitrarily defined biases from distributional word vectors

2. Use the augmented bias specification to create training examples for the explicit debiasing specialization net (DebiasNet)

DebiasNet model is a simple FFDN:

- Distributional vectors as input, specialized ("debiased") as output
- Loss: after specialization, targets from two lists equidistant to attributes

$$L_D = \left(\cos_d\left(\mathbf{t}'_1, \mathbf{a}'\right) - \cos_d\left(\mathbf{t}'_2, \mathbf{a}'\right)\right)^2$$

$$\mathbf{t}'_1 = \mathrm{DBN}(\mathbf{t}_1; \theta),\ \mathbf{t}'_2 = \mathrm{DBN}(\mathbf{t}_2; \theta),\ \text{and}\ \mathbf{a}' = \mathrm{DBN}(\mathbf{a}; \theta)$$

- Regularization: retaining the useful semantic information

$$L_R = \cos_d(\mathbf{t}_1, \mathbf{t}'_1) + \cos_d(\mathbf{t}_2, \mathbf{t}'_2) + \cos_d(\mathbf{a}, \mathbf{a}')$$

DebiasNet [Lauscher et al., 19]

DebiasNet is an direct/explicit specialization model for removing arbitrarily defined biases from distributional word vectors

- Good (explicit) debiasing performance
- Minor losses of general semantic quality (as measured on SimLex)

		WEAT T8 (gender bias, science vs. art)							WEAT T1 (sentiment, flowers vs. insects)						
		Explicit			Implicit		SemQ		Explicit			Implicit		SemQ	
	Model	WEAT	ECT	BAT	KM	SVM	SL	WS	WEAT	ECT	BAT	KM	SVM	SL	WS
FT	Distributional	1.30	73.5	41.0	100	100	38.2	73.8	1.67	46.2	56.1	95.7	100	38.2	73.0
	GBDD	0.96	84.7	33.9	62.9	**50.0**	**38.4**	**73.8**	0.08*	96.2	41.7	56.0	53.1	38.1	72.9
	BAM	0.10*	71.8	38.4	99.8	100	37.7	70.4	1.57	50.3	56.0	95.7	100	37.4	71.5
	DN	**0.05***	79.1	**33.6**	99.8	100	34.1	65.1	0.18*	79.8	45	95.7	100	35.09	68.6

Cross-Lingual (Transfer of) Semantic Specialization

(What to do for languages with no external resources/constraints?)

Semantic Specialization in Other Languages

The vast majority of specialization work presented so far fine-tunes the **English distributional vectors** using **lexico-semantic constraints available** for **EN**

How could we **transfer** the specialization to resource-poor languages?

Assumption:

- For most other languages **we do not have** comparable external resources
- We **cannot cheaply/trivially obtain** relevant constraints

But...didn't you just say that BabelNet covers 280+ languages?

- It does, but the number of covered concepts varies drastically across the langs
- What if we're specializing for some other relation (e.g., sentiment alignment)?

But...can't we create target-language constraints with some rules like in Morph-fitting?

- Constraints induced based on morphological rules work only for similarity-based spec.

Semantic Specialization in Other Languages

For **some relations** and **some languages** it **might** be possible to obtain language-specific relation-specific constraints cheaply

- Morph-fitting for similarity-based specialization
- LE constraints for LE-specialization via Hearst patterns

But **in general** this cannot be assumed

- We must think of methods for specializing target-language vectors without target-language constraints
- We do have the source language (typically EN) constraints and can specialize source language vectors
- Cross-lingual transfer: bridging the language chasm!

Cross-Lingual Transfer for Semantic Specialization

Two main paradigms for cross-lingual specialization transfer:

1. Cross-lingual distributional word vector space

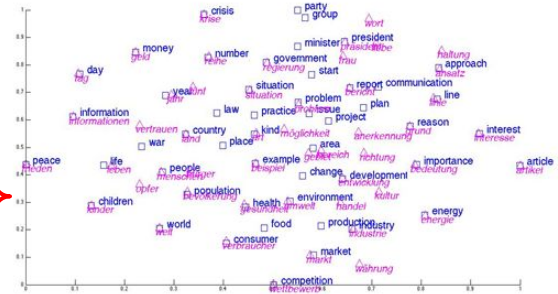


Image from [Luong et al., 2015]

2. Machine-translate source-language constraints to the target language

Specialization Transfer via CLWEs

Two strategies for transfering the specialization from the source language to the target language via the **shared representation space**:

1. **Joint specialization & transfer**
   - Word translation pairs between the languages as additional constraints
   - Cross-lingual alignment & semantic specialization in a single optimization
   - Attract-Repel [Mrkšić et al., TACL-17], CLEAR [Vulić et al., ACL-19]

2. **Transfer via non-specialized CLWE space**
   - Induce a bilingual non-specialized CLWE space
   - Learn the specialization function on the source-language subspace
   - Apply the learned spec. function to the target-language subspace
     [Vulić et al., NAACL-18; Glavaš & Vulić, ACL-18,19; Ponti et al., EMNLP-18]

Linguistic Constraints in Multiple Languages

	EN	DE	NL	SV	FR	ES	IT	PT	RU	PL	SH	BG
EN	64/1	25/ 1	31/2	21/1	36/2	32/2	36/2	26/1	120/1	28/1	19/1	14/1
DE	25/1	14/0	29/1	22/1	29/1	27/1	28/1	22/1	18/1	23/1	17/1	13/0
NL	31/2	29/1	68/3	24/1	35/2	32/2	34/2	26/1	20/1	28/1	20/1	15/1
SV	21/1	22/1	24/1	0/0	24/1	22/1	23/1	19/1	14/1	20/1	15/1	11/0
FR	36/2	28/1	35/2	24/1	103/1	36/2	39/2	28/1	22/1	31/1	21/1	15/1
ES	32/2	27/1	32/2	22/1	36/2	114/0	34/2	26/1	20/1	28/1	19/1	14/1
IT	35/2	28/1	34/2	23/1	39/2	34/2	16/1	28/2	22/1	30/2	21/1	15/1
PT	26/1	22/1	26/1	19/1	28/1	26/1	28/2	72/0	16/1	23/1	17/1	12/1
RU	20/1	18/1	20/1	14/1	22/1	20/1	22/1	16/1	5/0	17/1	12/1	9/0
PL	28/1	23/1	28/1	20/1	31/1	28/1	30/2	23/1	17/1	39/0	19/1	13/1
SH	19/1	17/1	20/1	15/1	21/1	19/1	21/1	16/1	12/1	19/1	0/0	10/1
BG	14/1	13/0	15/1	11/0	15/1	14/1	15/1	12/1	9/0	13/1	10/1	18/0

Linguistic constraint counts in tens of thousands...

Multilingual Attract-Repel [Mrkšić et al., TACL-19]

Cross-Lingual Vector Spaces

Constraint-based optimisation can be extended to cross-lingual specialisation. BABELNET constraints are used to bring the word vector spaces of various languages into a single unified vector space.

en_carpet			en_woman		
Slavic+EN	**Germanic**	**Romance+EN**	**Slavic + EN**	**Germanic**	**Romance+EN**
en_rug	de_teppichboden	en_rug	ru_женщина	de_frauen	fr_femme
bg_килим	nl_tapijten	it_moquette	bg_жените	sv_kvinnliga	en_womanish
ru_ковролин	en_rug	it_tappeti	sh_žena	sv_kvinna	es_mujer
bg_килими	de_teppich	pt_tapete	en_womanish	sv_kvinnor	pt_mulher
pl_dywany	en_carpeting	es_moqueta	bg_жена	de_weib	es_fémina
bg_мокет	de_teppiche	it_tappetino	pl_kobieta	en_womanish	en_womens
pl_dywanów	sv_mattor	en_carpeting	sh_treba	sv_kvinno	pt_feminina
sh_tepih	sv_matta	pt_carpete	bg_жени	de_frauenzimmer	pt_femininas
pl_wykładziny	en_carpets	pt_tapetes	en_womens	sv_honkön	es_femina
ru_ковер	nl_tapijt	fr_moquette	pl_kobiet	sv_kvinnan	fr_femelle
ru_коврик	nl_kleedje	en_carpets	sh_žene	nl_vrouw	pt_fêmea
sh_ćilim	nl_vloerbedekking	es_alfombra	pl_niewiasta	de_madam	fr_femmes
en_carpeting	de_brücke	es_alfombras	sh_žensko	sv_kvinnligt	it_donne
pl_dywan	de_matta	fr_tapis	sh_ženke	sv_gumman	es_mujeres
ru_ковров	nl_matta	pt_tapeçaria	pl_samica	sv_female	pt_fêmeas
en_carpets	en_mat	it_zerbino	ru_самка	sv_gumma	es_hembras

Joint Specialization & CL Transfer

(Multilingual) **Attract-Repel** [Mrkšić et al., TACL-17]

Use all available **monolingual** and **cross-lingual** constraints from all languages

1. Cross-lingual constraints amount to word translations, and they **align the distributional spaces between languages**

   Attract(en_car, de_Auto)
   Attract(en_car, it_macchina)

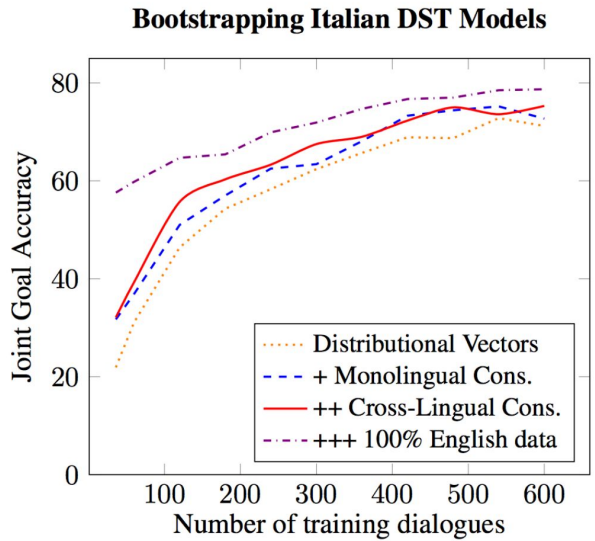
2. Monolingual constraints **drive the semantic specialization**

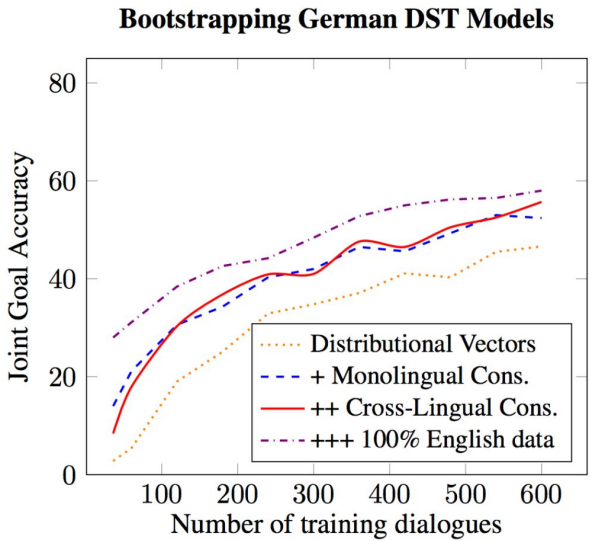
   Attract(en_car, en_automobile), *Attract(it_veloce, it_rapido)*
   Repel(en_black, en_white), *Repel(it_respettoso, it_irrispetoso)*

Language Understanding for Dialogue without Annotated Data?

Ontology Grounding: Multilingual DST Models

The domain ontology (i.e. the concepts it expresses) is language agnostic, which means that 'labels' persist across languages. Using training data for two (or more) languages, and cross-lingual vectors of high quality, we train the first-ever *multilingual* DST model.





[Mrkšić et al., TACL-19]

CLEAR: Cross-lingual LE Specialisation [Vulić et al., ACL-19]

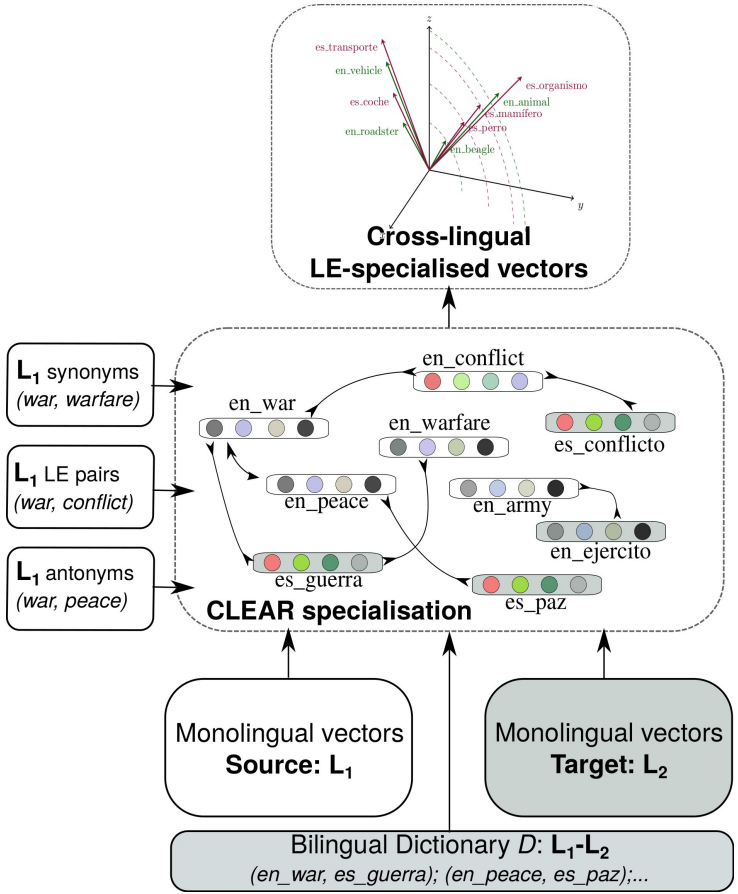
Joint Specialization & CL Transfer for LE

- Fine-tuning to reflect graded lexical entailment **in the target language** and **cross-lingually**

- **CLEAR = Cross-Lingual Lexical Entailment Attract-Repel**
  - Based on the monolingual LEAR method [Vulić and Mrkšić, 2018]

- Additional **translation objective** that enables the specialization **transfer**

$$Att_D(\mathcal{B}_D) = \lambda_D \sum_{k=1}^{K} \|\mathbf{x}_l^{(k)} - \mathbf{x}_r^{(k)}\|.$$



Cross-Lingual LE Evaluation: CL-HyperLex

- Point of departure: **English HyperLex** (2,616 word pairs)
  - Carefully designed sampling procedure; a variety of lexical relations represented

Step 1: Word Pair Translation

- Adopting the approach from similar work on cross-lingual and multilingual semantic similarity
  [Leviant and Reichart, 2015; Camacho-Collados et al., 2015, 2017]
- Two translators + another translator to resolve disagreement: translation agreement 85%-90% across the three target languages (DE, IT, HR)

(EN) conflit -- disagreement → (DE) Konflikt -- Uneinigkeit
(IT) conflitto -- disaccordo
(HR) sukob -- neslaganje

Step 2: Guidelines and Scoring

- Adopted from the original HyperLex
- The scoring interval is [0-6]; we recruited 5 native speaker annotators (for each target language)

CL-HyperLex: Construction Work

Step 3: Create Cross-Lingual Data Sets

- **Point of departure**: multilingual and cross-lingual semantic similarity datasets [Camacho Collados et al., 2015, 2017]

1. Intersecting aligned concept pairs...

2. ... and averaging their corresponding monolingual scores...

(EN) conflict -- disagreement 5.2
(DE) Konflikt -- Uneinigkeit 4.75

(EN-DE) conflict -- Uneinigkeit 4.975
(DE-EN) Konflikt -- disagreement 4.975

3. ...but we work only with pairs where the corresponding monolingual scores differ by 1.0 or less.

CL-HyperLex: Final Product

Examples

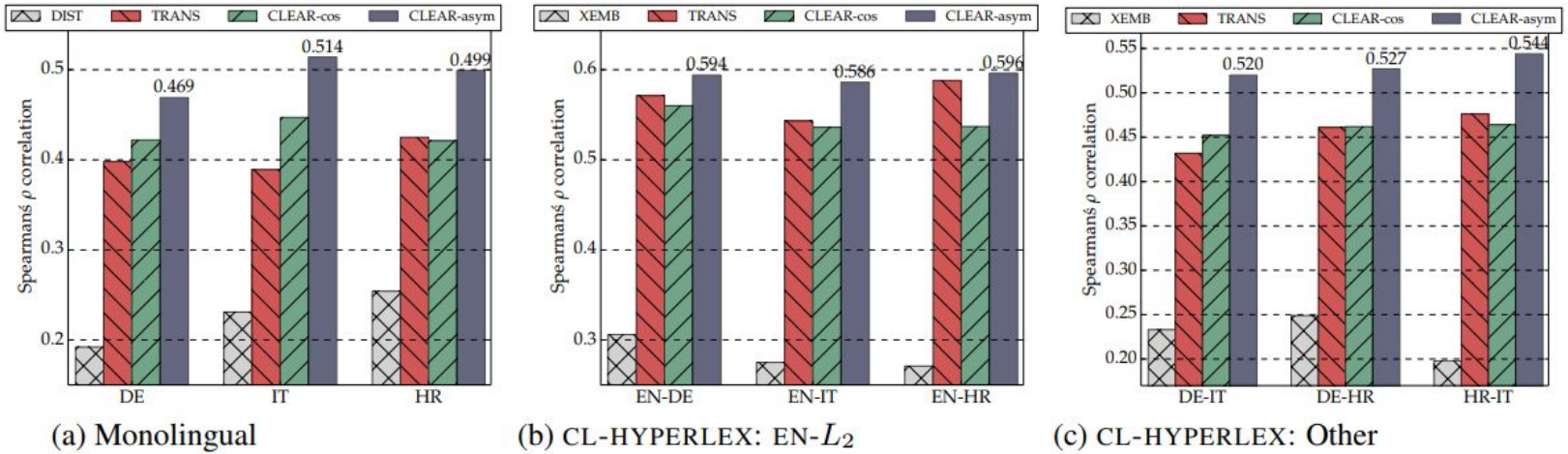
Monolingual Datasets			
EN	portrait	picture	5.90
DE	Idol	Person	4.0
DE	Motorrad	Fahrrad	0.25
IT	origano	cibo	3.25
HR	tenis	rekreacija	5.75
Cross-Lingual Datasets (CL-HYPERLEX)			
EN-DE	dinosaur	Kreatur	4.75
EN-IT	eye	viso	0.6
EN-HR	religija	belief	4.92
DE-IT	Medikation	trattamento	5.38
DE-HR	Form	prizma	0.0
IT-HR	aritmetica	matematika	5.5

Size

	EN	DE	IT	HR
EN	2,616	3,029	3,338	3,514
DE	–	2,616	3,424	3,522
IT	–	–	2,616	3,671
HR	–	–	–	2,616

Results: CLEAR on Graded LE (CL-Hyper)

- Baselines:
  - DIST: Non-specialized space (distributional vectors)
  - XEMB: Cross-lingual word embeddings without fine-tuning for LE
  - TRANS: LEAR specialization of EN space + translating DE/IT/HR words to EN for inference
- Results on CL-HyperLex



(a) Monolingual (b) CL-HYPERLEX: EN-L_2 (c) CL-HYPERLEX: Other

Commercial Break: SemEval 2020 Task 2

Shared Task on Predicting Cross-Lingual and Multilingual (Graded) LE

https://competitions.codalab.org/competitions/20865

Predict (grade and binary) LE:

- **Monolingually**: EN, DE, IT, HR TR, + surprise language
- **Cross-lingually**: all 15 possible language pairs of the above 6 languages

Two tracks:

- Fully unsupervised models (use only unannotated corpora)
- Externally-informed models (any kind of resource is allowed)
  - E.g., WordNet, BabelNet, ...

Transfer via Non-Specialized Cross-Lingual Space

Cross-lingual alignment as a pre-processing step before specialization

1. **Induce CLWEs from monolingual distributional spaces**

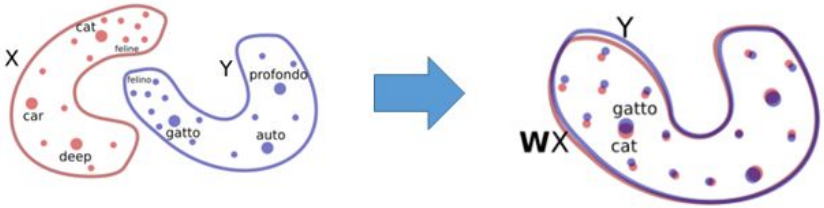


Image from [Conneau et al., ICLR-18]

- This (in most cases) means *learn the (linear) function* ***g*** that projects the target language distributional space **Y** to the source language space **X**

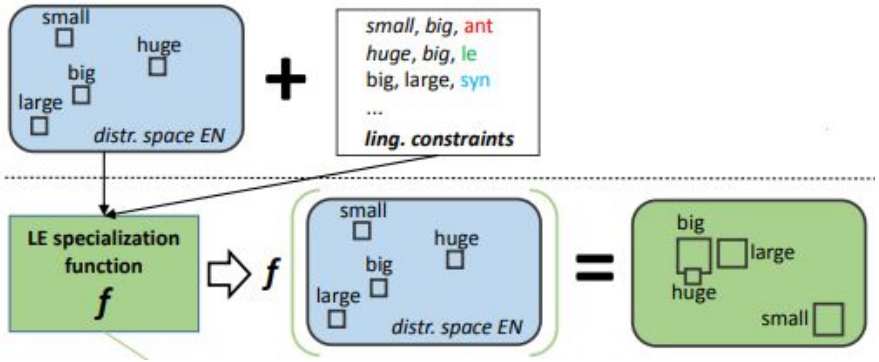
- Joint bilingual space: **X U *g*(Y)**

Transfer via Non-Specialized Cross-Lingual Space

Cross-lingual alignment as a pre-processing step before specialization

2. Learn the specialization function using source language constraints

- Specialization function ***f*** learned on source dist. space **X** using source language constraints (e.g., with *post-specialization* or *explicit specialization*)



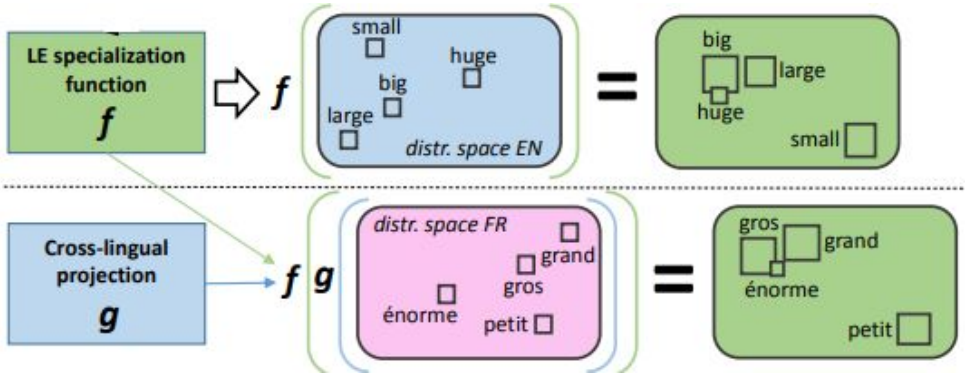
- Specialized source space: **X'** = ***f*(X)**

Transfer via Non-Specialized Cross-Lingual Space

Cross-lingual alignment as a pre-processing step before specialization

3. Apply the specialization function to (projected) target language vectors

- Specialization function ***f*** applied to the target language vectors, previously projected to the source space, ***g(Y)***



- Specialized target lang. space: **Y' = *f*(*g*(Y))**

Quick Detour:
Cross-Lingual Word Embeddings

(Key mechanism for cross-lingual transfer of specialization models, and other NLP models)

Inducing a Cross-Lingual Distributional Space

Different methodologies but the same **end goal**:

Induce a semantic vector space in which words with similar meaning end up with similar vectors, whether they come from the same language or from different languages.

Typology of methods for inducing CLWEs [Ruder et al., '18]

1. **Type of bilingual / multilingual signal**
   - Document-level, sentence-level, word-level, no signal (i.e., unsupervised).
2. **Comparability**
   - Parallel texts, comparable texts, not comparable (i.e., randomly aligned)
3. **Point (time) of alignment**
   - Joint embedding models vs. Post-hoc alignment
4. **Modality**
   - Text only vs. using images for alignment (e.g., [Kiela et al., '15])

Joint CLWE models

Jointly learning embeddings of two or more languages from scratch

1. **Using word translations**
   - Shared vectors for words in translation pairs [Guo et al., ‘14]
     - Feeding contexts from both languages to a standard embedding model (e.g., Skip-Gram)
   - Creation of pseudo-bilingual corpus

     [Gouws & Søgaard, ’15; Ammar et al., ’15; Duong et al., ’16; Adams et al., ‘17]
     - Pseudo-bilingual corpus by replacing words in a monolingual corpus with their translations

2. **Using sentence translations**
   - Compositional sentence model [Hermann & Blunsom, ‘13]
   - Bilingual Skip-Gram [Gouws et al., ‘15; Luong et al., ‘15]

Projection-Based CLWEs

Post-hoc alignment of independently trained monolingual word vector spaces

- Alignment based on **word translation pairs** (dictionary **D**)
- Supervised models use pre-obtained **D**, unsupervised automatically induce **D**

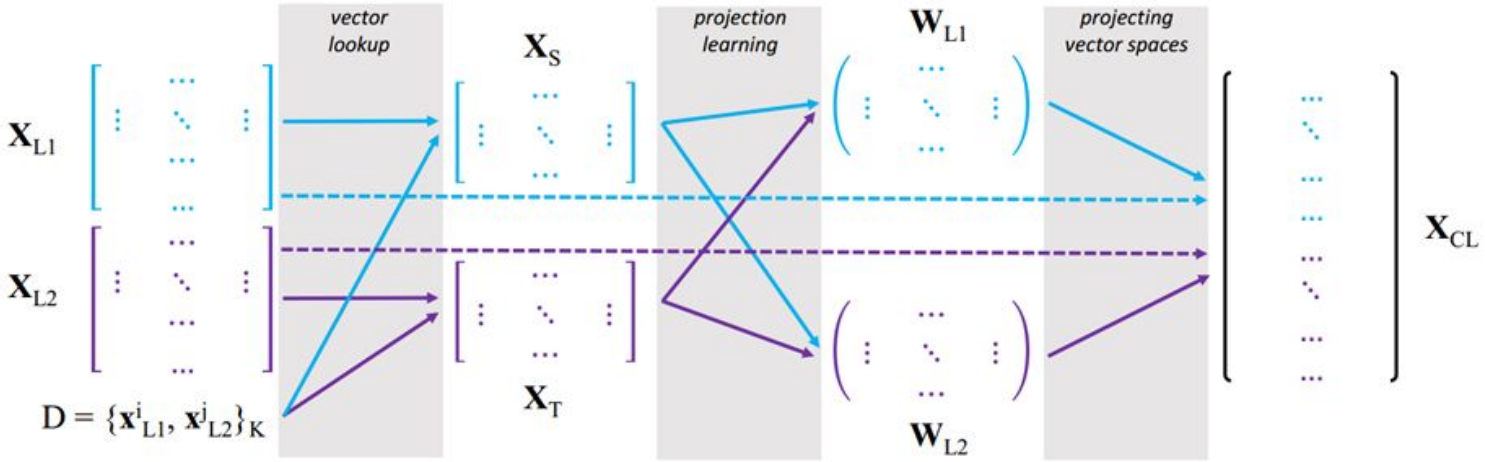


Image from [Glavaš et al., ACL-19]

Projection-based CLWEs

Most models learn a single projection matrix $\mathbf{W}_{L1}$ (i.e., $\mathbf{W}_{L2} = \mathbf{I}$)

$$\begin{array}{r} \\ \text{bird} \\ \text{pretty} \\ \dots \\ \text{eat} \end{array} \overset{\mathbf{X}_S}{\begin{bmatrix} -1.18 & 0.21 & \dots & 0.11 \\ 0.23 & -0.53 & \dots & 0.34 \\ \dots & \dots & \dots & \dots \\ 0.78 & 1.33 & \dots & -0.47 \end{bmatrix}} \cdot \mathbf{M} = \overset{\mathbf{X}_T}{\begin{bmatrix} 0.59 & 1.01 & \dots & 0.37 \\ -0.34 & -0.27 & \dots & 0.41 \\ \dots & \dots & \dots & \dots \\ 0.81 & -0.31 & \dots & 0.29 \end{bmatrix}} \begin{array}{l} \\ \text{Vogel} \\ \text{schön} \\ \dots \\ \text{essen} \end{array}$$

How do we find the „optimal" projection matrix $\mathbf{W}_{L1}$?

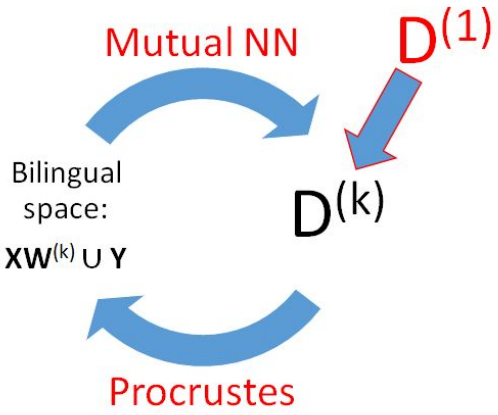
- Mean square distance [Mikolov et al., '13] (and all subsequent work), except
- (Relaxed) Cross-Domain Similarity Local Scaling [Joulin et al., '18]

Projection-based CLWEs

But...we still need some **cross-lingual word-level supervision/signal** between the **source language** and the **target language**, right?

Unsupervised induction of CLWEs: different approaches for inducing the initial dictionary $\mathbf{D}^{(1)}$, e.g.:

- Adversarial learning [Conneau et al., '18]
- Similarities of similarity distribution [Artetxe et al., 2018]
- PCA [Hoshen & Wolf, '18]
- Solving optimal transport problem [Alvarez & Jaakkola, '18]



Assumption of (approximate) isomorphism of monolingual

And Why Do We Need Fully Unsupervised Induction of CLWEs?

Motivation for CLWEs in general

- **Simple**: quick and efficient to train
- (Still) **state-of-the-art** in cross-lingual NLP
- **Light-weight** and **inexpensive**
- Multilingual modeling of meaning and **supporting cross-lingual transfer** for downstream NLP tasks

Motivation for Unsupervised CLWEs models:

- Wide portability without bilingual resources?
- NLP models for virtually any language?
- Increasing the ability of cross-lingual transfer?
- Better performance?
- Or just theoretically and conceptually attractive without true empirical justification?

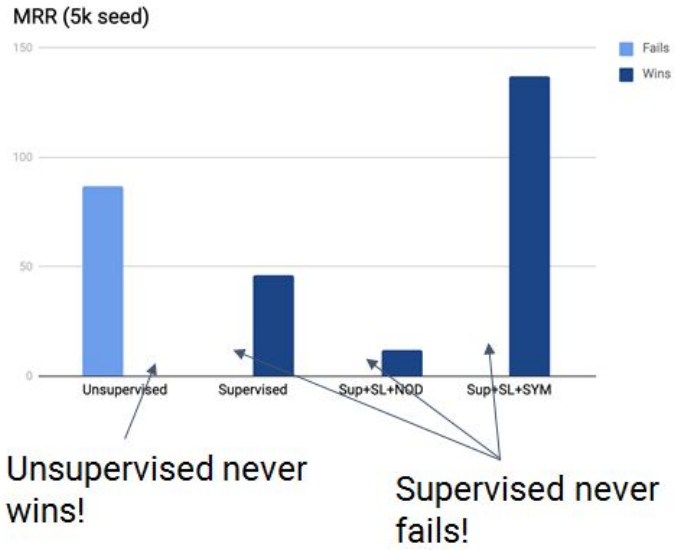
(Weakly) Supervised vs. Unsupervised Induction of CLWEs

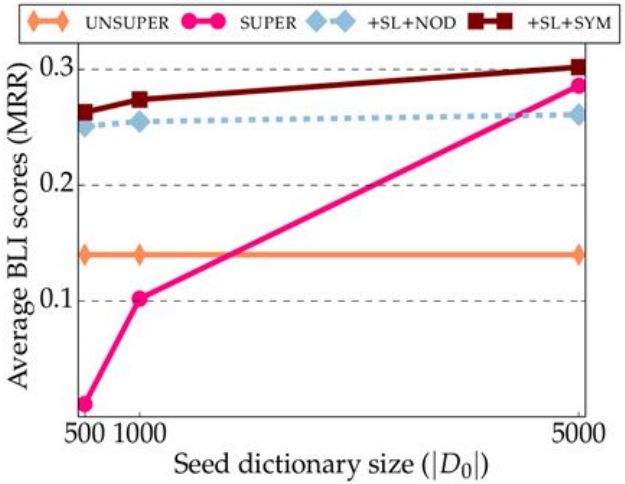
Findings from recent comparative analyses
[Glavaš et al., ACL-19; Vulić et al., EMNLP-19]

1. Unsupervised CLWE induction methods **cannot outperform** (weakly) supervised counterparts
2. Unsupervised CLWE methods **fail** to induce meaningful CLWE spaces:
    - Distant and typologically different languages
    - Monolingual embeddings induced from non-comparable corpora
        - E.g., L1 vectors induced from Wikipedia, L2 from EuroParl

(Weakly) Supervised vs. Unsupervised Induction of CLWEs

Thu, Nov 7, Session 9A: *Do We Really Need Fully Unsupervised Cross-Lingual Embeddings?* Ivan Vulić, Goran Glavaš, Roi Reichart and Anna Korhonen





Back to Specialization Transfer (Based on CLWEs)

(Key mechanism for cross-lingual transfer of specialization models, and other NLP models)

Specialization Transfer via CLWEs

Semantic similarity

- Post-specialization [Ponti et al., EMNLP-18]
- Explicit specialization [Glavaš & Vulić, ACL-18]

Lexical entailment (hypernymy)

- POSTLE (post-spec. for LE) [Kamath et al., RepL4NLP-19]
- GLEN (explicit spec. for LE) [Glavaš & Vulić, ACL-19]
- BiSparse-Dep [Upadhyay et al., NAACL-18]

Other relations

- Debiasing CL-transfer [Lauscher et al., 19]
- Induction of VerbNet classes [Vulić et al., EMNLP-17]

Cross-Lingual Transfer for Similarity Specialization

Adversarial specialization CL transfer [Ponti et al., EMNLP-18]

- Intrinsic evaluation: {DE, IT}-SimLex-999
- Downstream evaluation: DST (DE, IT), Lexical Simplification (IT)

	Similarity (ρ)		LS (Acc)		DST (JGA)	
Vector space	IT	DE	IT	DE	IT	DE
Distrib.	.297	.417	.308	-	.681	.621
AUXGAN	**.431**	**.525**	**.392**	-	**.714**	**.651**

Explicit specialization CL transfer [Glavaš & Vulić, ACL-18]

- Intrinsic evaluation: {DE, IT, HR}-SimLex-999

Model	German	Italian	Croatian
Distributional (X)	.407	.360	.249
ER-Specialized (X′)			
ER-MSD	.415	.406	.287
ER-CNT	**.533**	**.448**	**.315**

Cross-Lingual Transfer for LE Specialization

(Adversarial) post-specialization for Lexical Entailment [Kamath et al., RepL4NLP-19]

LE detection (binary classification), CL transfer for:

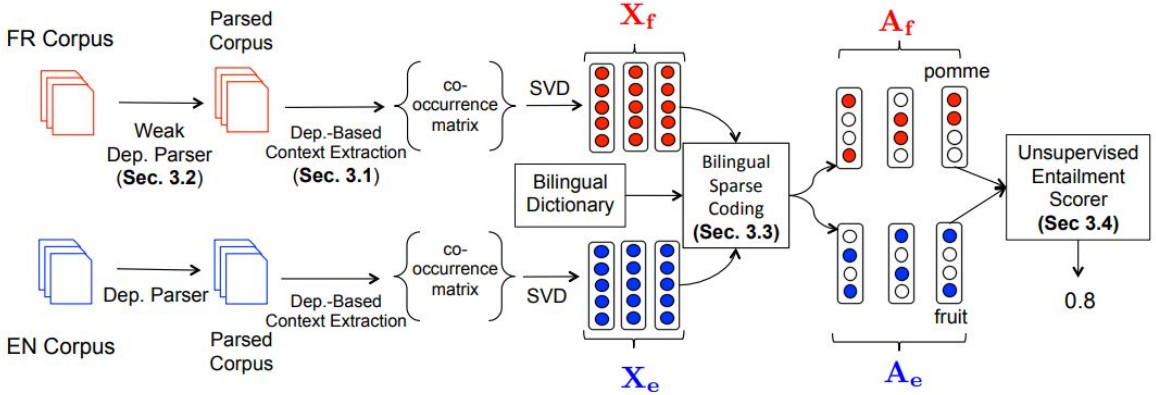
- FR and ES
- Using three different methods for inducing shared spaces
  - Sup. Procrustes [Smith et al., ICLR-17]
  - Unsup. MUSE [Conneau et al., ICLR-18]
  - Unsup. VecMap [Artetxe et al., ACL-18]

	Target: SPANISH			Target: FRENCH		
Random	.498			.515		
Distributional	.362			.387		
	Ar	Co	Sm	Ar	Co	Sm
POSTLE DFFN	**.798**	.740	.728	.688	.735	.742
POSTLE ADV	.768	**.790**	**.782**	**.746**	**.770**	**.786**

Cross-Lingual Transfer for LE Specialization

BiSparse-Dep model [Upadhyay et al., NAACL-18]

1. Sparse dependency-based distributional vectors for 2 languages
2. Monolingual embeddings: SVD of sparse dep. dist. vectors
3. Alignment (bilingual space induction) via constrained (dictionary) optimization
4. Unsupervised entailment scoring function [Kotlerman et al., ACL-09], based on the distributional inclusion hypothesis [Geffet & Dagan, ACL-05]

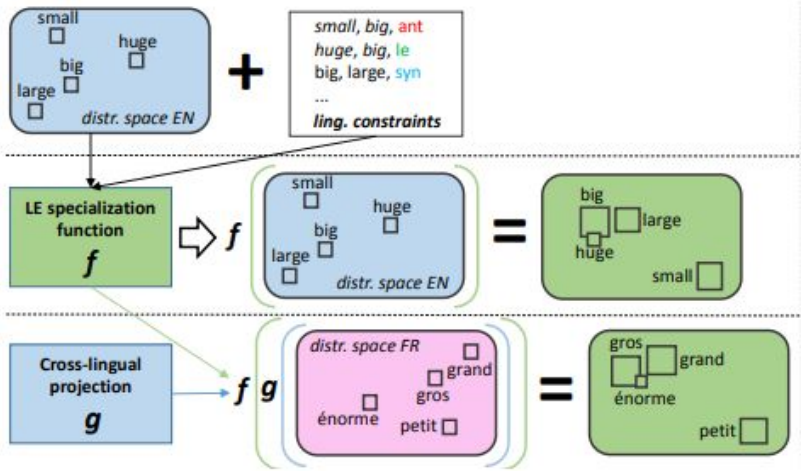


Cross-Lingual Transfer for LE Specialization

GLEN (explicit specialization for LE)
[Glavaš & Vulić, ACL-19]

Evaluation on cross-lingual LE detection datasets
[Upadhyay et al., NAACL-18]

- EN-AR, EN-RU, and EN-FR
- HYPO = LE vs. inverse LE
- COHYP = LE vs. co-hyponymy



	Model	EN-FR	EN-RU	EN-AR	Avg
HYPO	CL-DEP	.538	.602	.567	.569
	BI-SPARSE	.566	.590	.526	.561
	GLEN	**.792**	**.811**	**.816**	**.806**
COHYP	CL-DEP	.610	.562	.631	.601
	BI-SPARSE	.667	.636	.668	.657
	GLEN	**.779**	**.849**	**.821**	**.816**

Cross-Lingual Transfer for Semantic Specialization

Two main paradigms for cross-lingual specialization transfer:

1. Cross-lingual distributional word vector space

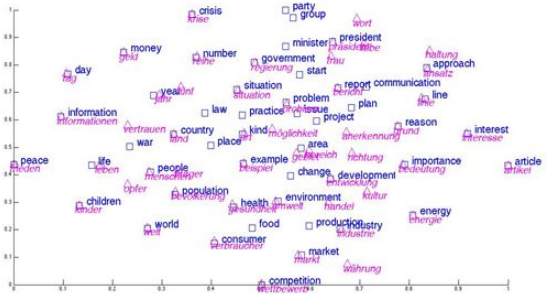
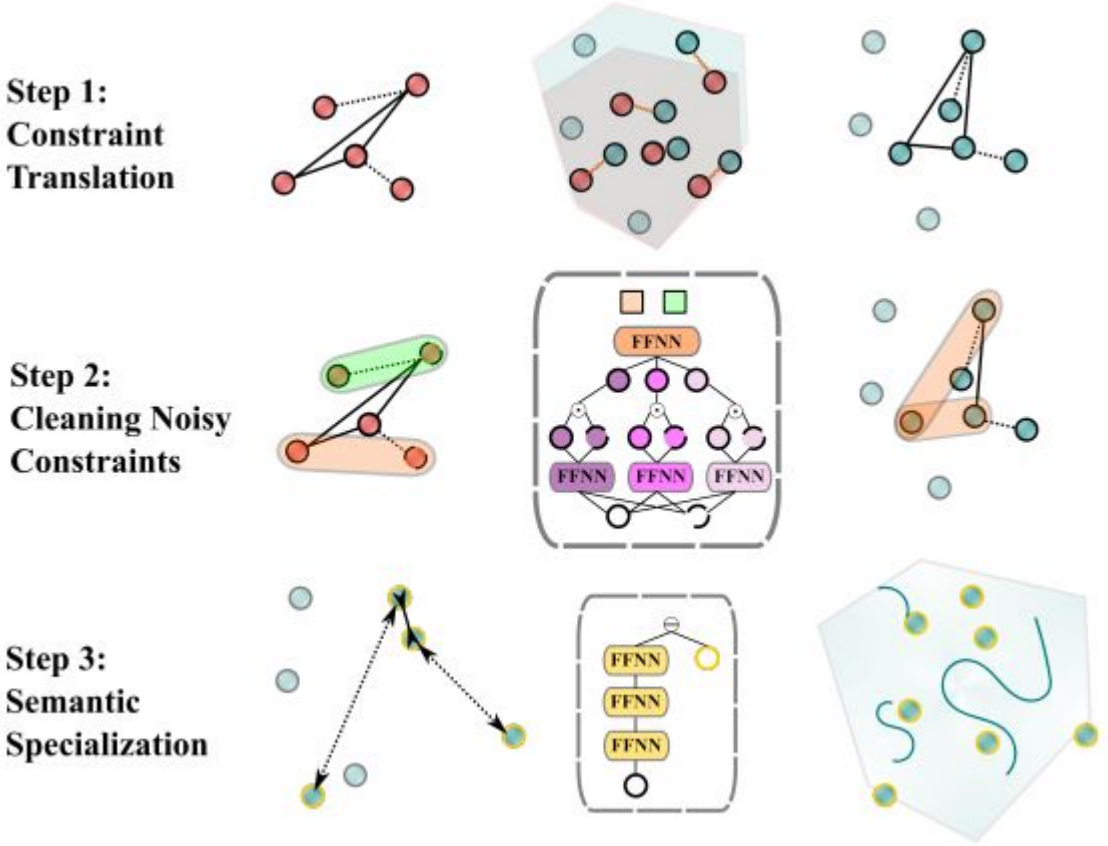


Image from [Luong et al., 2015]

2. Machine-translate source-language constraints to the target language

Cross-lingual Specialization via Relation Induction [Ponti et al., EMNLP-19]



Cross-lingual Specialization via Relation Induction [Ponti et al., EMNLP-19]

Step 1. Constraint Translation

Attract & Repel constraints in the source language: (w^l_S, w^r_S)

Translated, noisily, to the target language via a bilingual CL space: (w^l_T, w^r_T)

- w^l_T nearest neighbour of w^l_S, w^l_T nearest neighbour of w^l_S

Step 2. Cleaning Noisy Constraints

Lexical relation classification, modified Specialization Tensor Model [Glavaš & Vulić, NAACL-18]

$$\text{FFN}^{\sigma} \bigoplus_{i=1}^{k} \left\{ \text{FFN}^{\tau}_i(\mathbf{x}_l)^{\top} W_i \, \text{FFN}^{\tau}_i(\mathbf{x}_r) + \mathbf{b}_i \right\}$$

One *Attract* classifier, one *Repel*

Trained on source lang. vector pairs $(\mathbf{x}^l_S, \mathbf{x}^r_S)$

Predicts on target lang. vector pairs $(\mathbf{x}^l_S, \mathbf{x}^r_S)$

Cross-lingual Specialization via Relation Induction [Ponti et al., EMNLP-19]

Step 3. Adversarial Post-Specialization Using Clean Constraints

AR-based adversarial post-specialization model [Ponti et al., EMNLP-18]

Evaluation:

- Intrinsic: Word similarity for 8 languages
- Downstream: DST (DE, IT), Lex. simp. (IT, ES, PT), STS (AR)

Baseline: cross-lingual transfer of the specialization function directly via the shared bilingual space (**X-Postspec**)

Cross-lingual Specialization via Relation Induction [Ponti et al., EMNLP-19]

Model	DE	IT	HE	FI	HR	TR	PL	RU
Distributional	.426	.304	.368	.240	.344	.535	.395	.270
X-PS	.503	.392	.380	.314	.376	.464	.344	.402
CLSRI-AR	.500	**.525**	.454	.394	.425	.554	.433	.331
CLSRI-PS	**.565**	.512	**.522**	**.490**	**.505**	**.613**	**.534**	**.507**

Table 1: Spearman's ρ correlation scores for 8 languages on datasets for intrinsic evaluation of true semantic similarity. The models in comparison are briefly summarized in § 4 and in Figure 1.

Model	DE	IT
Distributional	0.640	0.729
X-Postspec	0.647	0.737
CI-AR	0.652	0.745
CI-Postspec	**0.687**	**0.782**

Table 2: Joint goal accuracy scores in the DST task.

	LS			STS
Model	IT	ES	PT	AR
Distributional	0.28	0.27	0.27	0.67
X-PS	0.38	0.57	0.55	0.66
CLSRI-AR	0.35	0.51	0.33	0.66
CLSRI-PS	**0.51**	**0.74**	**0.72**	**0.70**

Table 3: Lexical Simplification (LS) performance for Italian, Spanish, and Portuguese; Semantic Textual Similarity (STS) performance for Arabic.

Conclusion:

Constraints translation + monolingual spec. >> specialization function transfer

Specialization of Contextualized Embeddings

Table of Contents

Contextualized embeddings as **transfer learning**

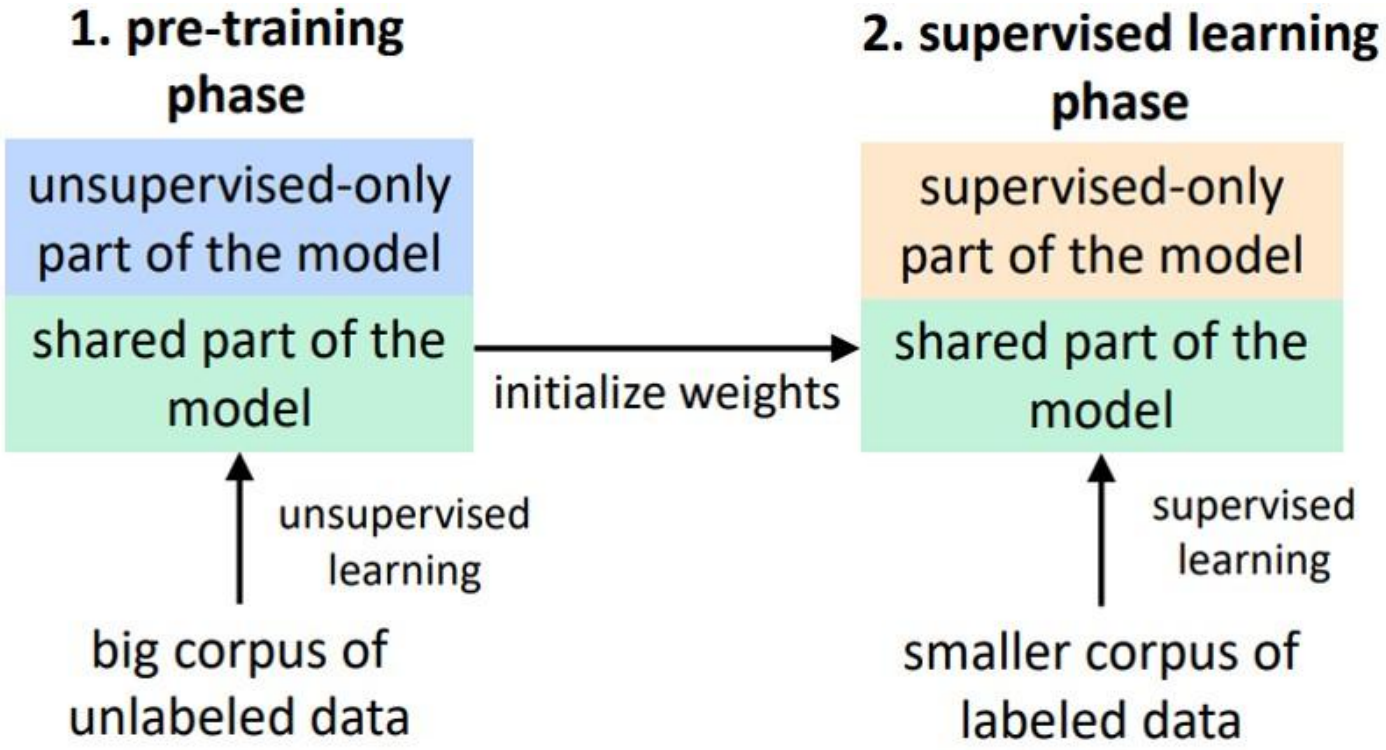
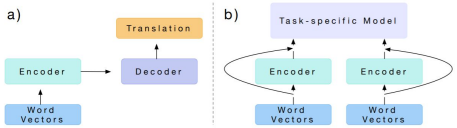


Image credits: https://helicqin.github.io

A dichotomic tree of CWEs

Pretraining task?

Machine Translation

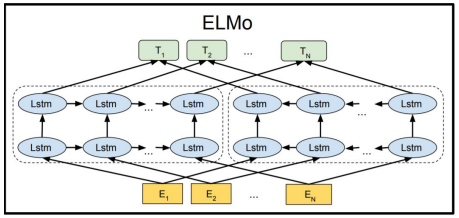


CoVe (McCann et al. 2017)

Language Modeling

Neural Architecture?

Bi-LSTM

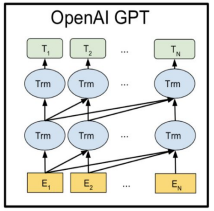


ELMo (Peters et al. 2019)

Transformer

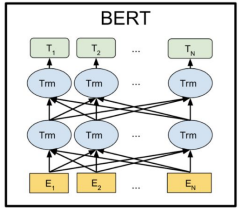
Context?

Mono-



GPT (Radford et al. 2018)

Bi-directional

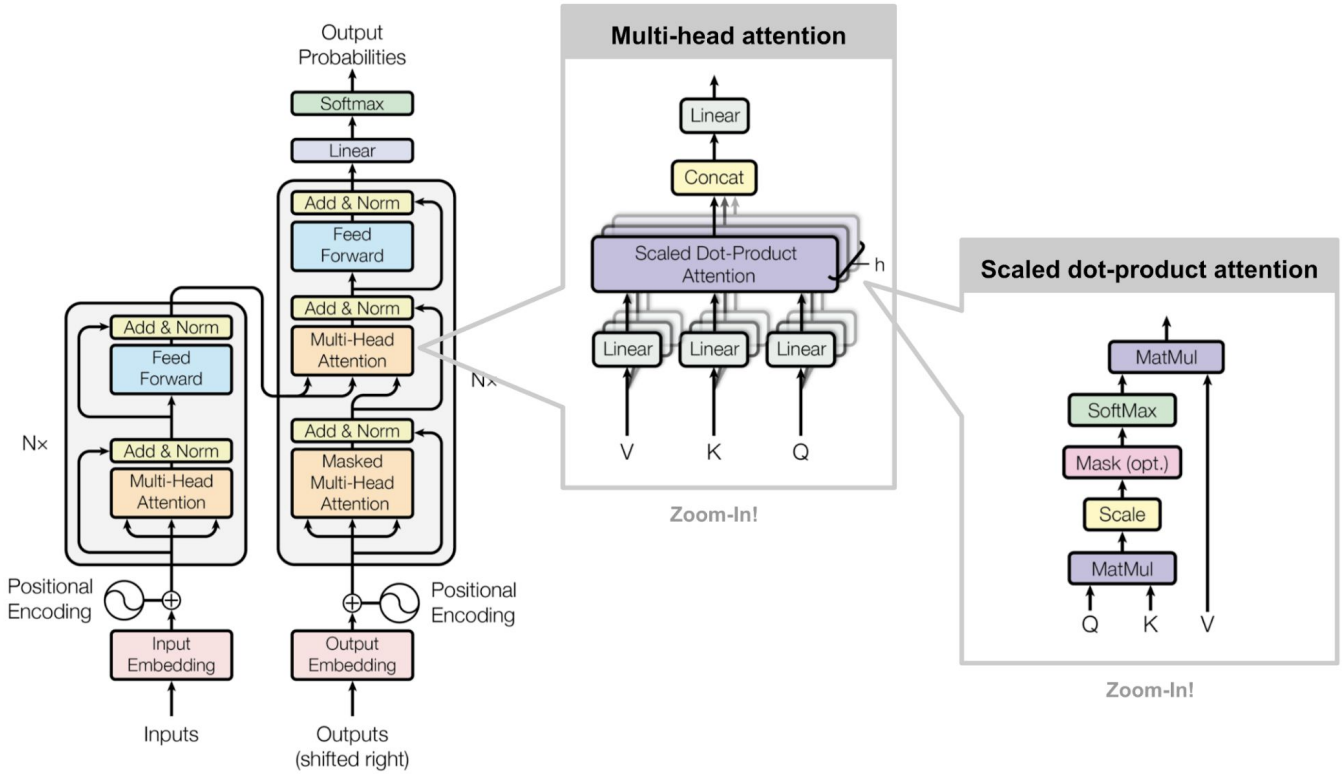


BERT (Devlin et al. 2019)

Key Concepts

The encoder during supervised learning	is frozen (**feature-based**)	is trainable (**fine-tuning**)
Token representations from	a **weighted sum of all** layers	the **last** layer
Sentence representation from	the **last time step**	a **special BOS token**
The classifier for supervised learning	is a **full-fledged architecture**	is a **linear layer**
Example	ELMo	BERT

Dissecting BERT*



All the specialization methods for CWEs thus far are based on BERT

Is specialization obsolete for CWEs?

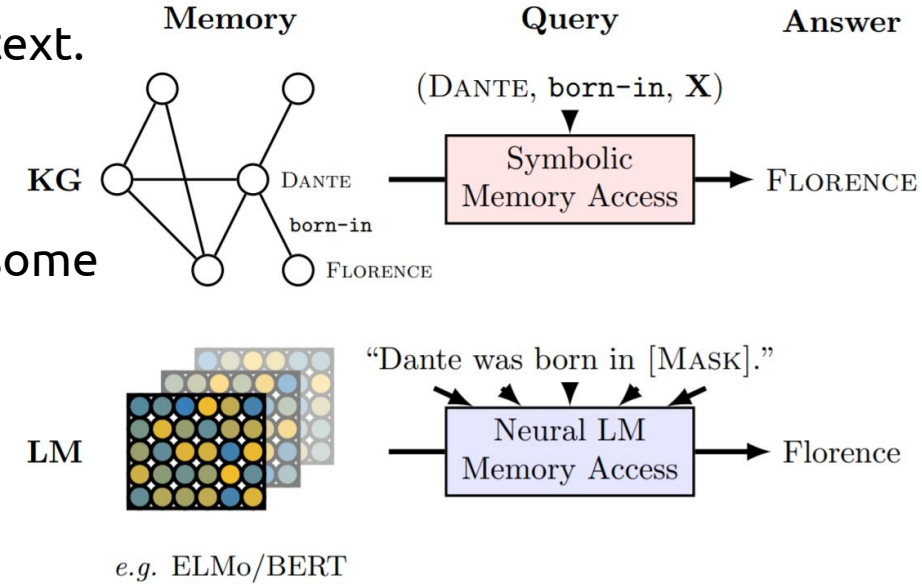
Linguistic specialization: 👌

Paradigmatic relations not found in context.

World knowledge specialization: ❓

Language models capture KB triples to some extent. See Petroni et al. (2019):

- Performance comparable with supervised models (P@10=57.1)
- Unreliable for N-to-M relations



github.com/facebookresearch/LAMA

New Wine, New Bottles

Some traditional specialization methods for static embeddings, such as Attract-Repel, **cannot be applied directly** to CWEs.

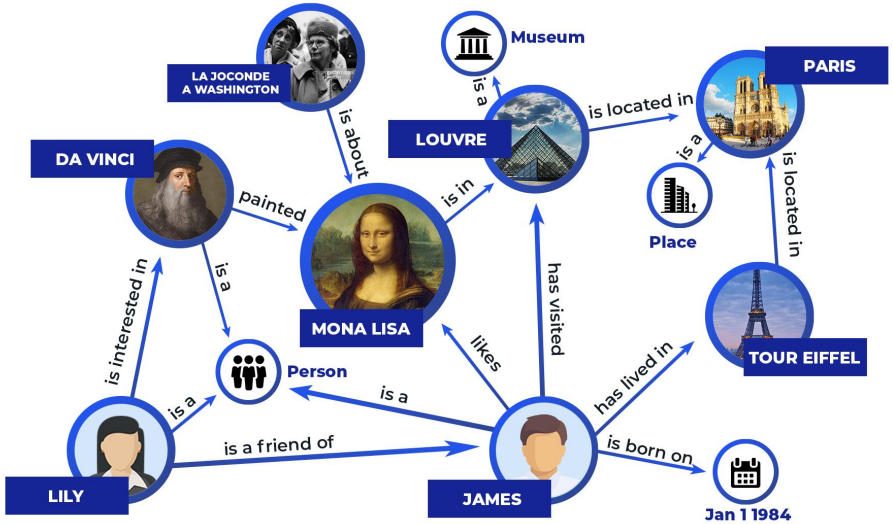
They are predicated on the assumption of a static space **W**, which is perturbed according to some constraints.

CWEs are a function of **W** and $\boldsymbol{\theta}_{\text{encoder}}$ so there is no explicit space to perturb.

Moreover, pure post-processing may incur in **catastrophic forgetting** of the pre-training task.

Challenges for CWE specialization

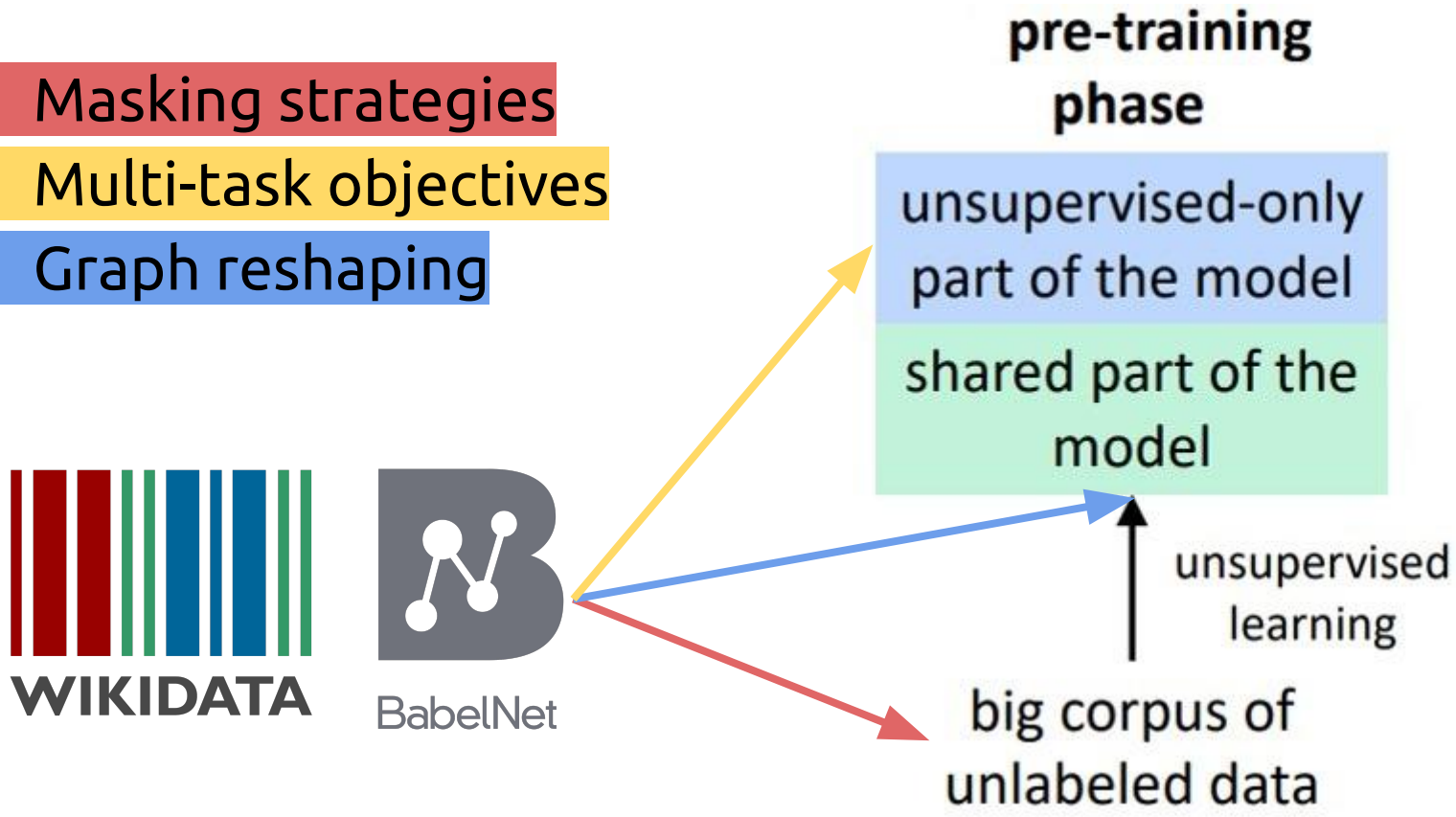
"The *Mona Lisa* has an enigmatic expression."



- **Heterogeneity** of the inputs: graphs vs linear sequences.
- **Asymmetry** between pre-training (texts linked to graphs available) and supervised learning phase.
- **Knowledge Noise**: Irrelevant facts may divert the meaning from the original sentence.

A (Big)Bird's-eye view of specialization methods

- Masking strategies
- Multi-task objectives
- Graph reshaping



A (Big)Bird's-eye view of specialization methods

Specialization method	Model	Notes
Masking strategies	Baidu-ERNIE (Sun et al. 2019a)	
Multi-task objectives	LIBERT (Lauscher et al. 2019)	Relation Prediction, external knowledge
	THU-ERNIE (Zhang et al. 2019)	Denoising Auto-Encoding, external knowledge
	Baidu-ERNIE 2.0 (Sun et al. 2019b)	Token- and Sentence-level Classification, multiple un-/weakly supervised tasks
Graph reshaping	K-BERT (Liu et al. 2019)	

Table of Contents

Masking Strategies

Pretraining models learn the interaction between masked elements and the context → masking should be applied to all **units of meaning**.

Model	**Targets of masking strategy**
BERT (Devlin et al. 2019)	WordPieces (word segments)
BERT-WWM (Cui et al. 2019)	Whole words
SpanBert (Joshi et al. 2019)	Contiguous random spans
RoBERTa (Liu et al. 2019)	Dynamycally changed WordPieces

Baidu-ERNIE (Sun et al. 2019a)

Higher-level masking strategies, in addition to token masking:

1. **Masking of phrases** identified through chunking.
2. **Masking of entities**, identified through Named Entity Recognition.

Implicit specialization, as the model learns the relation between entities and other entities / events in context.

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

Code: https://github.com/PaddlePaddle/ERNIE/tree/develop/ERNIE

The importance of being ERNIE

Ablation study on the masking strategies on 10% of the raw texts, evaluated on Chinese XNLI: **both phrase- and entity-based masking are beneficial**.

mask strategy	dev Accuracy	test Accuracy
word-level(chinese character)	77.7%	76.8%
word-level&phrase-level	78.3%	77.3%
word-level&phrase-leve&entity-level	78.7%	77.6%

Baidu-ERNIE: Cloze test

No	Text	Predict by ERNIE	Predict by BERT	Answer
1	2006年9月，_____与张柏芝结婚，两人婚后育有两儿子——大儿子Lucas谢振轩，小儿子Quintus谢振南；	谢霆锋	谢振轩	谢霆锋
	In September 2006, _____ married Cecilia Cheung. They had two sons, the older one is Zhenxuan Xie and the younger one is Zhennan Xie.	Tingfeng Xie	Zhenxuan Xie	Tingfeng Xie
2	戊戌变法，又称百日维新，是_____、梁启超等维新派人士通过光绪帝进行的一场资产阶级改良。	康有为	孙世昌	康有为
	The Reform Movement of 1898, also known as the Hundred-Day Reform, was a bourgeois reform carried out by the reformists such as ____ and Qichao Liang through Emperor Guangxu.	Youwei Kang	Shichang Sun	Youwei Kang
3	高血糖则是由于_____分泌缺陷或其生物作用受损，或两者兼有引起。糖尿病时长期存在的高血糖，导致各种组织，特别是眼、肾、心脏、血管、神经的慢性损害、功能障碍。	胰岛素	糖糖内	胰岛素
	Hyperglycemia is caused by defective _____ secretion or impaired biological function, or both. Long-term hyperglycemia in diabetes leads to chronic damage and dysfunction of various tissues, especially eyes, kidneys, heart, blood vessels and nerves.	Insulin	(Not a word in Chinese)	Insulin
4	澳大利亚是一个高度发达的资本主义国家，首都为_____。作为南半球经济最发达的国家和全球第12大经济体、全球第四大农产品出口国，其也是多种矿产出口量全球第一的国家。	墨尔本	墨悉本	堪培拉
	Australia is a highly developed capitalist country with _____ as its capital. As the most developed country in the Southern Hemisphere, the 12th largest economy in the world and the fourth largest exporter of agricultural products in the world, it is also the world's largest exporter of various minerals.	Melbourne	(Not a city name)	Canberra (the capital of Australia)
5	_____是中国神魔小说的经典之作，达到了古代长篇浪漫主义小说的巅峰，与《三国演义》《水浒传》《红楼梦》并称为中国古典四大名著。	西游记	《小》	西游记
	_____is a classic novel of Chinese gods and demons, which reaching the peak of ancient Romantic novels. It is also known as the four classical works of China with Romance of the Three Kingdoms, Water Margin and Dream of Red Mansions.	The Journey to the West	(Not a word in Chinese)	The Journey to the West
6	相对论是关于时空和引力的理论，主要由_____创立。	爱因斯坦	卡尔斯所	爱因斯坦
	Relativity is a theory about space-time and gravity, which was founded by _____.	Einstein	(Not a word in Chinese)	Einstein

BERT:

1) **copies other NEs** in the context

2, 5) gets the **correct type but the wrong entity**

3, 4, 6) predicts **ill-formed words**

Baidu-ERNIE:

All correct except 4

Table of Contents

LIBERT (Lauscher et al. 2019)

Linguistically Informed BERT

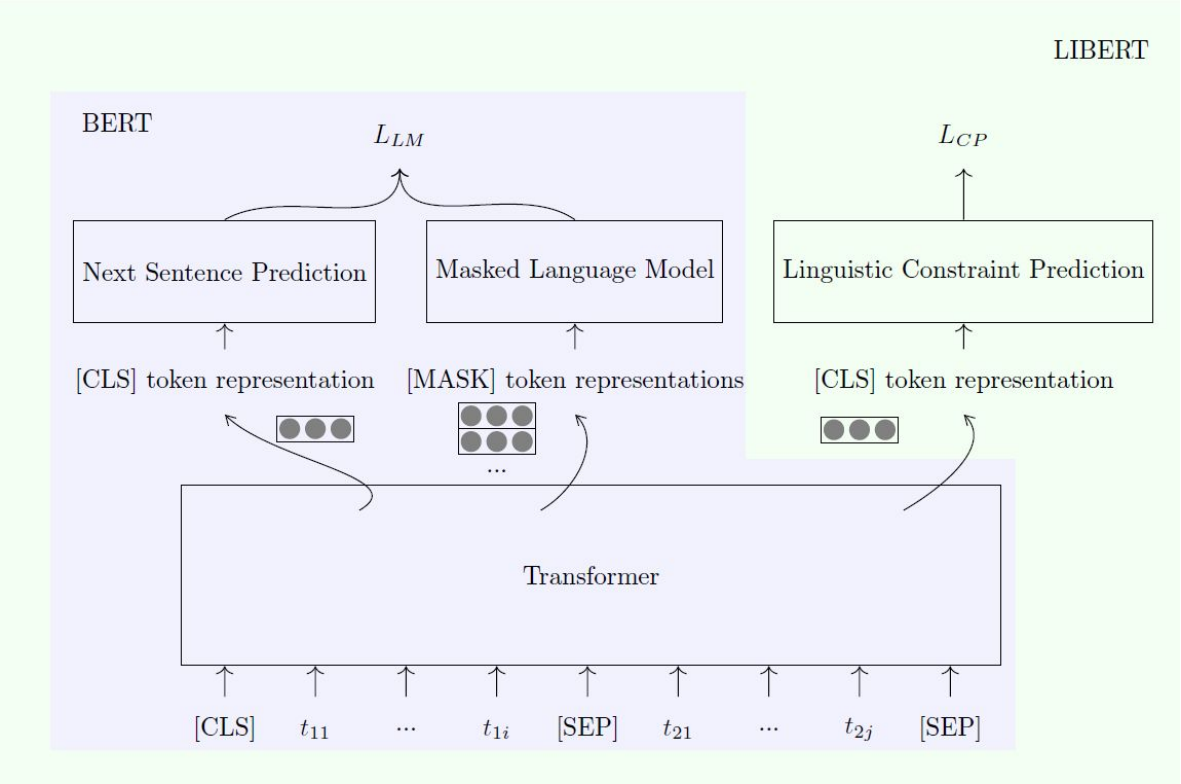
Positive examples of constraints $C = \{(w_1, w_2)_i\}_{i=1}^{N}$ are sourced from lexical resources (WordNet and Roget's Thesaurus).

Negative examples are obtained by substituting each member of a positive pair with the nearest neighbour among the words in the same batch, based on an auxiliary static embedding space.

Each constraint is then transformed into a BERT-compatible input, e.g.

[CLS]	men	#ded	[SEP]	reg	#ener	#ated	[SEP]
0	0	0	0	1	1	1	1

LIBERT: Architecture



Binary single-layer classifier

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{x}_{CLS}\mathbf{W}_{LRC}^{\top} + \mathbf{b}_{LRC})$$

with a negative log-likelihood loss:

$$L_{CP} = -\sum_{k} \ln \hat{\mathbf{y}}_k \cdot \mathbf{y}_k$$

LIBERT: Results

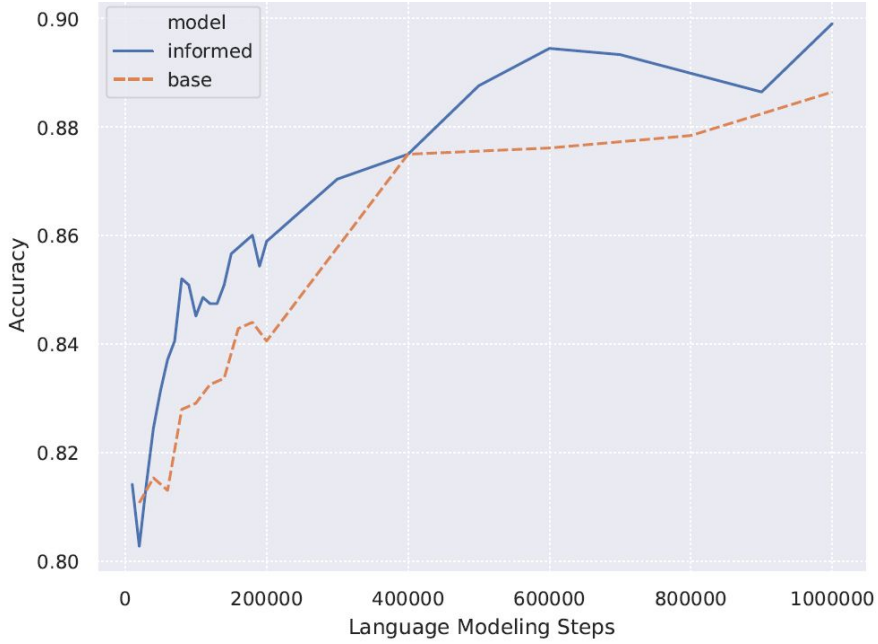
Better performance in 8/10 tasks, especially in the *Linguistic Acceptability* and *Diagnostic* datasets.

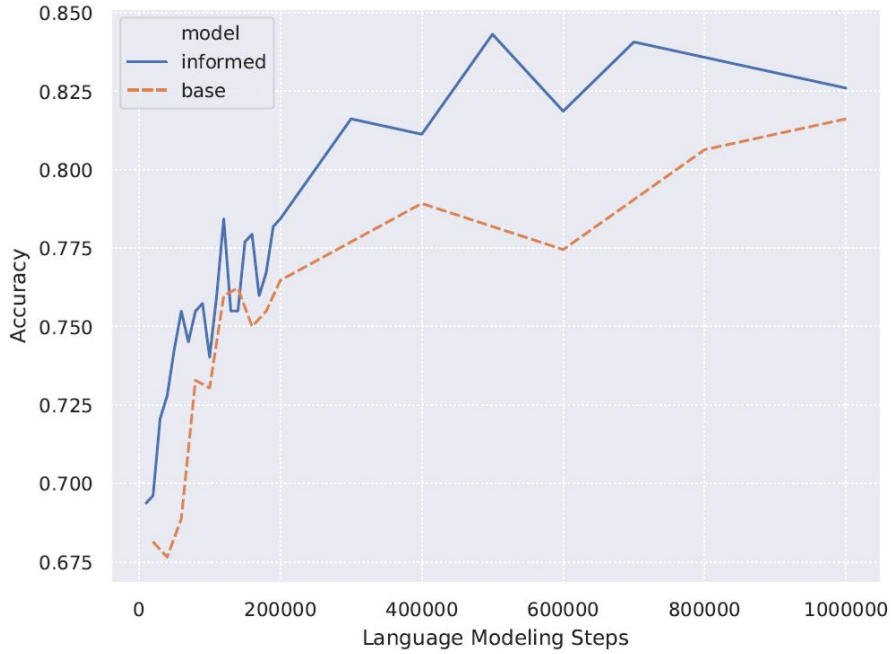
		CoLA MCC	SST-2 Acc	MRPC F1/Acc	STS-B Pears/Spearm	QQP F1/Acc
Dev	BERT	29.4	88.7	87.1/81.6	86.4/73.3	85.9/89.5
	LIBERT	**35.3**	**89.9**	**87.9/82.6**	**87.2/75.6**	**86.3/89.8**
	Δ	+5.9	+1,2	+0.8/+1.0	+0.8/2.3	+0.4/+0.3
Test	BERT	21.5	87.9	84.8/78.8	**80.8/79.3**	68.6/87.9
	LIBERT	**31.4**	**89.6**	**86.1/80.4**	80.5/78.8	**69.0/88.1**
	Δ	+9.9	+1.7	+1.3/+1.6	-0.3/-0.5	+0.4/+0.2

		MNLI-m Acc	MNLI-mm Acc	QNLI Acc	RTE Acc	AX MCC
Dev	BERT	78.2	**78.8**	86.2	63.9	–
	LIBERT	**78.5**	78.7	**86.5**	**65.3**	–
	Δ	+0.3	-0.1	+0.3	+1.4	–
Test	BERT	78.2	**77.6**	85.8	61.3	26.3
	LIBERT	**78.4**	77.4	**86.2**	**62.6**	**32.8**
	Δ	+0.2	-0.2	+0.4	+1.3	+6.5

Benefit over Time

SST-2 (left) and MRPC (right) evaluations for LIBERT (Lauscher et al. 2019) vs a BERT baseline. **The benefit of specialization does not fade away over time!**





THU-ERNIE (Zhang et al. 2019)

THU-ERNIE introduces a novel objective, **Denoising Auto-Encoding for entities**

During both pre-training and refinement, entities are labelled via NER.

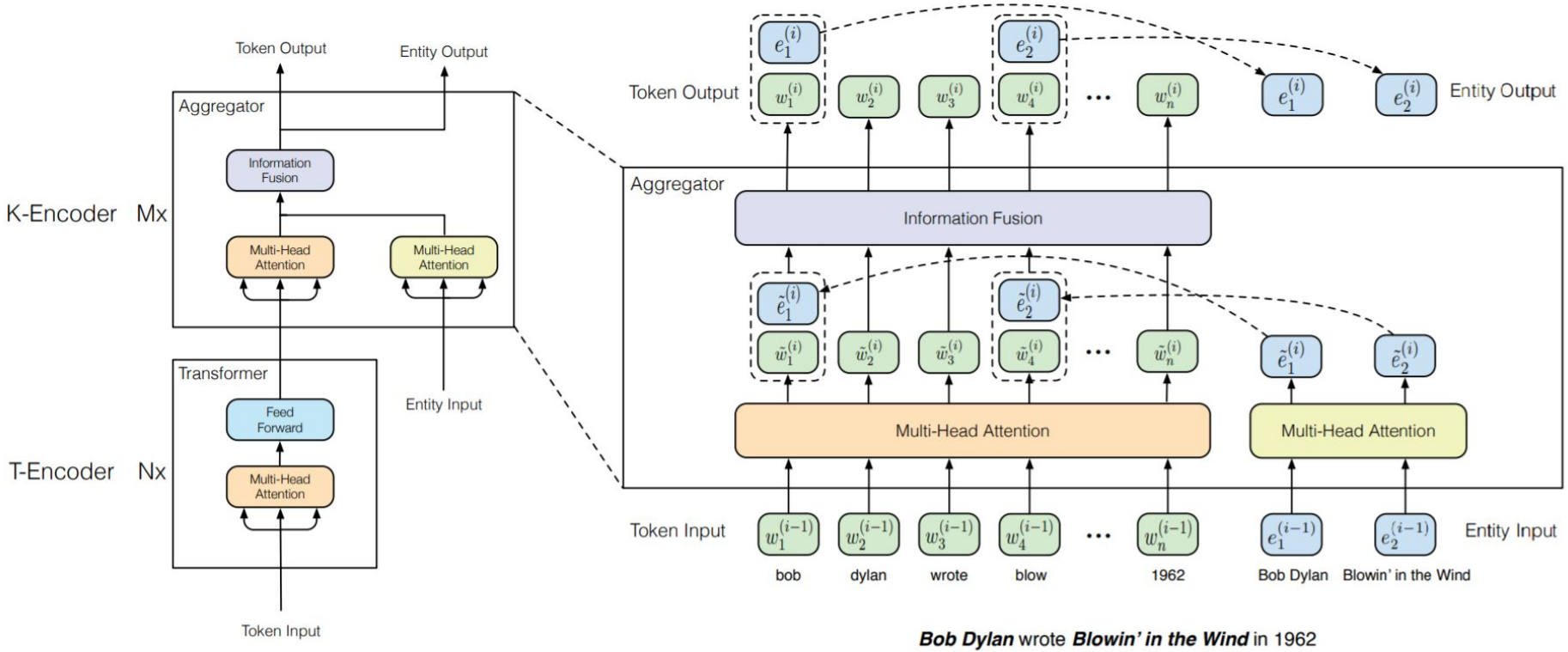
The model comprises:

1. **Transformer layers** for word tokens (T-Encoder)
2. **Aggregator layers** for word-entity fusion (K-Encoder).

Input words are represented as **CWEs** (output of the T-Encoder).

Input entities are **static embeddings** trained on Wikidata via TransE.

THU-ERNIE: Architecture



(a) Model Achitecture

(b) Aggregator

THU-ERNIE: Aggregator

Input words **w** and input entities **e** are first encoded separately through a Multi-Headed attention mechanism:

$$\{\tilde{\boldsymbol{w}}_1^{(i)}, \ldots, \tilde{\boldsymbol{w}}_n^{(i)}\} = \texttt{MH-ATT}(\{\boldsymbol{w}_1^{(i-1)}, \ldots, \boldsymbol{w}_n^{(i-1)}\}),$$
$$\{\tilde{\boldsymbol{e}}_1^{(i)}, \ldots, \tilde{\boldsymbol{e}}_m^{(i)}\} = \texttt{MH-ATT}(\{\boldsymbol{e}_1^{(i-1)}, \ldots, \boldsymbol{e}_m^{(i-1)}\}).$$

If a word is aligned to an entity (left), they are mutually integrated into a fused representation, and then mapped back to separate ones. Otherwise (right):

$$\boldsymbol{h}_j = \sigma(\tilde{\boldsymbol{W}}_t^{(i)} \tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{W}}_e^{(i)} \tilde{\boldsymbol{e}}_k^{(i)} + \tilde{\boldsymbol{b}}^{(i)}),$$
$$\boldsymbol{w}_j^{(i)} = \sigma(\boldsymbol{W}_t^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}),$$
$$\boldsymbol{e}_k^{(i)} = \sigma(\boldsymbol{W}_e^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_e^{(i)}).$$

$$\boldsymbol{h}_j = \sigma(\tilde{\boldsymbol{W}}_t^{(i)} \tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{b}}^{(i)}),$$
$$\boldsymbol{w}_j^{(i)} = \sigma(\boldsymbol{W}_t^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}).$$

THU-ERNIE: Entity Classifier

The entity representations for prediction are the **e** outputs of the last layer.

The classifier is simply a **softmax layer**:

$$p(e_j|w_i) = \frac{\exp(\texttt{linear}(\boldsymbol{w}_i^o) \cdot \boldsymbol{e}_j)}{\sum_{k=1}^{m} \exp(\texttt{linear}(\boldsymbol{w}_i^o) \cdot \boldsymbol{e}_k)},$$

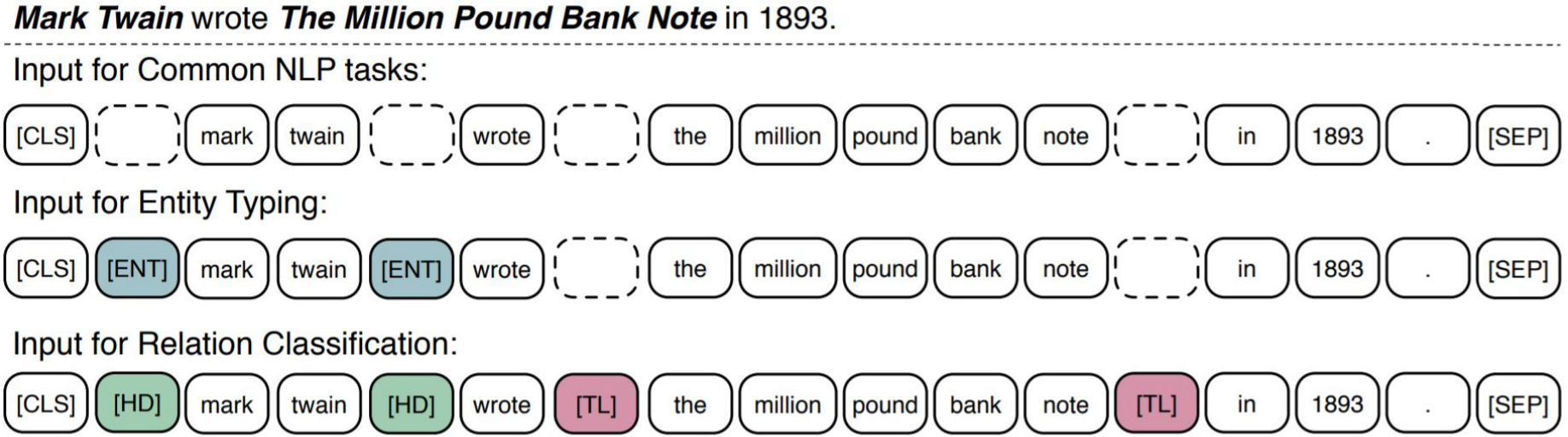
Candidate entities for prediction are selected among those in the **same batch**.

Noise injection for entities: 5% randomly replaced, 15% masked, 80% preserved.

Pre-processing for knowledge-driven tasks

Knowledge driven tasks: **relation prediction** and **entity typing**

New **special tokens** for these tasks: [HD] and [TL] for head entities and tail entities, [ENT] for entity mentions



THU-ERNIE: Results, good news...

Results for **Entity typing** (above) and **Relation Classification** (below)

Model	P	R	F1
NFGEC (LSTM)	68.80	53.30	60.10
UFET	77.40	60.60	68.00
BERT	76.37	70.96	73.56
ERNIE	**78.42**	**72.90**	**75.56**

FIGER

Model	Acc.	Macro	Micro
NFGEC (Attentive)	54.53	74.76	71.58
NFGEC (LSTM)	55.60	75.15	71.73
BERT	52.04	75.16	71.63
ERNIE	**57.19**	**76.51**	**73.39**

Open Entity

Noisy labels

Few-shot learning

Model	FewRel			TACRED		
	P	R	F1	P	R	F1
CNN	69.51	69.64	69.35	70.30	54.20	61.20
PA-LSTM	-	-	-	65.70	64.50	65.10
C-GCN	-	-	-	69.90	63.30	66.40
BERT	85.05	85.11	84.89	67.23	64.81	66.00
ERNIE	88.49	88.44	**88.32**	69.97	66.08	**67.97**

...and bad news

Results for **Natural Language Understanding** (GLUE benchmark)
comparable with baseline BERT

Model	MNLI-(m/mm) 392k	QQP 363k	QNLI 104k	SST-2 67k
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	-	93.5
ERNIE	84.0/83.2	71.2	91.3	93.5

Model	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k
$\text{BERT}_{\text{BASE}}$	52.1	85.8	88.9	66.4
ERNIE	52.3	83.2	88.2	68.8

Baidu-ERNIE 2.0 (Sun et al. 2019b)

In addition to external knowledge, auxiliary objectives can also take advantage of **textual information that is not distributional** in nature.

Continuous learning from unsupervised or weakly supervised tasks related to several levels of linguistic structure (lexicon, syntax, semantics).

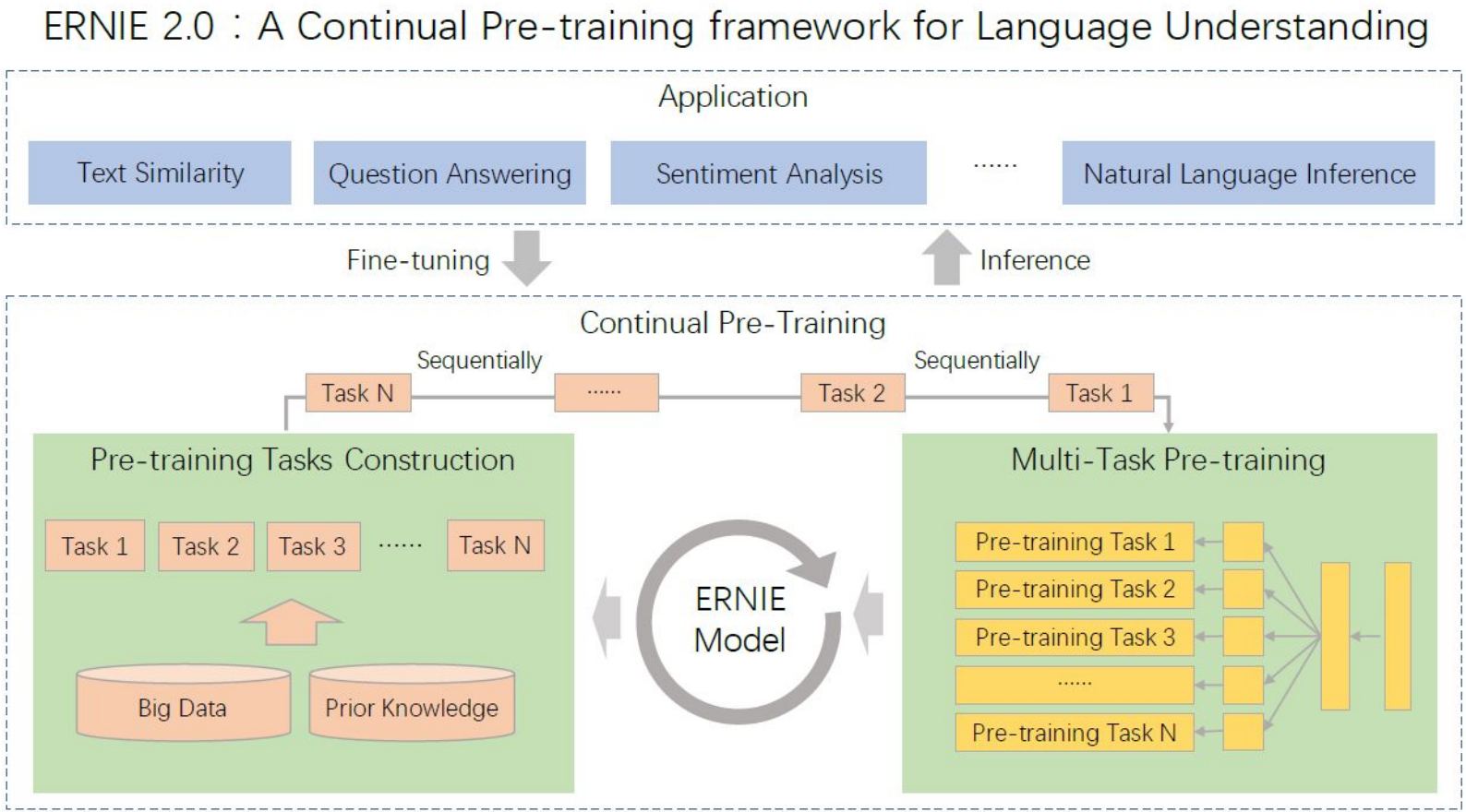
Iterative routine for pre-training:

1. Add new task to the objective.
2. Reach convergence.
3. Initialize new model with optimal parameters.
4. Start over from point 1.

Previously optimized parts are kept to **avoid catastrophic forgetting**.

Code: https://github.com/PaddlePaddle/ERNIE

Baidu-ERNIE 2.0: Framework



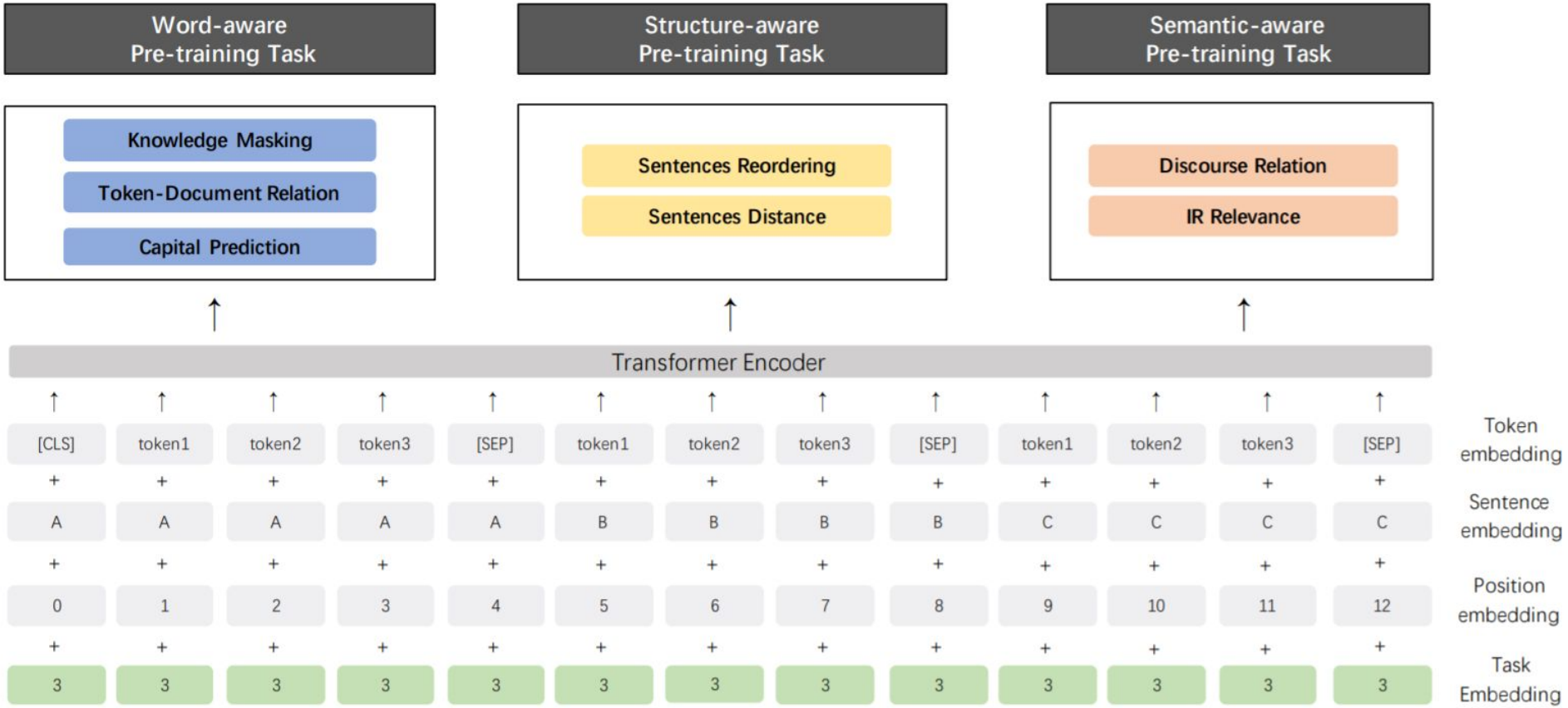
Pre-training tasks

1. Word-aware tasks
    a. **Knowledge Masking Task**: entity masking (cfr Baidu-ERNIE 1.0).
    b. **Capitalization Prediction Task**: was a token originally upper-case?
    c. **Token-Document Relation Prediction**: does a token appear in another segment from the same document? The goal is recognising if a token is a keyword for a topics.

2. Structure-aware tasks
    a. **Sentence Reordering**: k-class classification where k is the number of possible permutations.
    b. **Sentence Distance**: 3-class (two sentences are: adjacent, in the same document, unrelated).

3. Semantic-aware tasks
    a. **Discourse Relation**: relations are mined automatically through discourse markers.
    b. **IR Relevance**: 3-class. given a query and a title from the Baidu Search Engine log, is the title: clicked by the user, appearing among the results, or unrelated.

Baidu-ERNIE 2.0: Model



Baidu-ERNIE 2.0: Results

Task(Metrics)	*BASE model*		*LARGE model*				
	Test		Dev			Test	
	BERT	ERNIE 2.0	BERT	XLNet	ERNIE 2.0	BERT	ERNIE 2.0
CoLA (Matthew Corr.)	52.1	**55.2**	60.6	63.6	**65.4**	60.5	**63.5**
SST-2 (Accuracy)	93.5	**95.0**	93.2	95.6	**96.0**	94.9	**95.6**
MRPC (Accurary/F1)	84.8/88.9	**86.1/89.9**	88.0/-	89.2/-	**89.7/-**	85.4/89.3	**87.4/90.2**
STS-B (Pearson Corr./Spearman Corr.)	87.1/85.8	**87.6/86.5**	90.0/-	91.8/-	**92.3/-**	87.6/86.5	**91.2/90.6**
QQP (Accuracy/F1)	89.2/71.2	**89.8/73.2**	91.3/-	91.8/-	**92.5/-**	89.3/72.1	**90.1/73.8**
MNLI-m/mm (Accuracy)	84.6/83.4	**86.1/85.5**	86.6/-	**89.8/-**	89.1/-	86.7/85.9	**88.7/88.8**
QNLI (Accuracy)	90.5	**92.9**	92.3	93.9	**94.3**	92.7	**94.6**
RTE (Accuracy)	66.4	**74.8**	70.4	83.8	**85.2**	70.1	**80.2**
WNLI (Accuracy)	**65.1**	**65.1**	-	-	-	65.1	**67.8**
AX(Matthew Corr.)	34.2	**37.4**	-	-	-	39.6	**48.0**
Score	78.3	**80.6**	-	-	-	80.5	**83.6**

Task	Metrics	BERT$_{BASE}$		ERNIE 1.0$_{BASE}$		ERNIE 2.0$_{BASE}$		ERNIE 2.0$_{LARGE}$	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
CMRC 2018	EM/F1	66.3/85.9	-	65.1/85.1	-	69.1/88.6	-	**71.5/89.9**	-
DRCD	EM/F1	85.7/91.6	84.9/90.9	84.6/90.9	84.0/90.5	88.5/93.8	88.0/93.4	**89.7/94.7**	**89.0/94.2**
DuReader	EM/F1	59.5/73.1	-	57.9/72.1	-	61.3/74.9	-	**64.2/77.3**	-
MSRA-NER	F1	94.0	92.6	95.0	93.8	95.2	93.8	**96.3**	**95.0**
XNLI	Accuracy	78.1	77.2	79.9	78.4	81.2	79.7	**82.6**	**81.0**
ChnSentiCorp	Accuracy	94.6	94.3	95.2	95.4	95.7	95.5	**96.1**	**95.8**
LCQMC	Accuracy	88.8	87.0	89.7	87.4	**90.9**	**87.9**	**90.9**	**87.9**
BQ Corpus	Accuracy	85.9	84.8	86.1	84.8	86.4	85.0	**86.5**	**85.2**
NLPCC-DBQA	MRR/F1	94.7/80.7	94.6/80.8	95.0/82.3	95.1/82.7	95.7/84.7	95.7/85.3	**95.9/85.3**	**95.8/85.8**

Table of Contents

K-BERT (Liu et al. 2019)

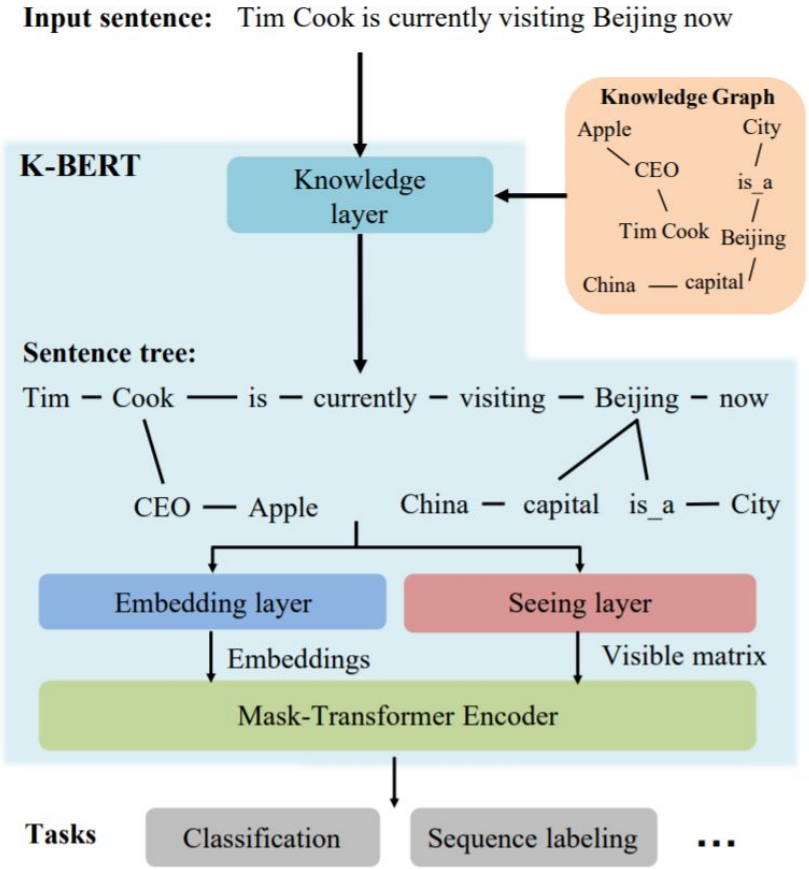
Contrary to other methods, K-BERT comes into play in the **refinement phase / inference**, not the pre-training phase → scalability

Core idea: **a sentence is a (degenerate) graph** where consecutive words are connected. Relevant triples from a KG can be connected to entity mentions

The resulting sentence tree is reshaped into linear form: the structural information is preserved through **soft position embeddings** and a **visible matrix**

The BERT architecture and objectives are otherwise unchanged.

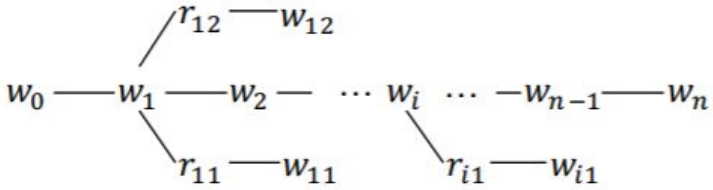
K-BERT: Architecture



Knowledge Layer

A **KG is queried** with all the entities appearing in a sentence.

The returned **triples are stitched** to the linear sequence, forming a **sentence tree** with max depth = 1.



K-BERT: Embedding Layer and Visible Matrix

Embedding Layer

Converts the sentence tree into an input representation summing:

1) **token embeddings**: tokens in the branch are inserted after the entity nodes
2) **soft position embeddings**: distance in edges from BOS (Note: not unique)
3) **segment embeddings**.

Visible Matrix

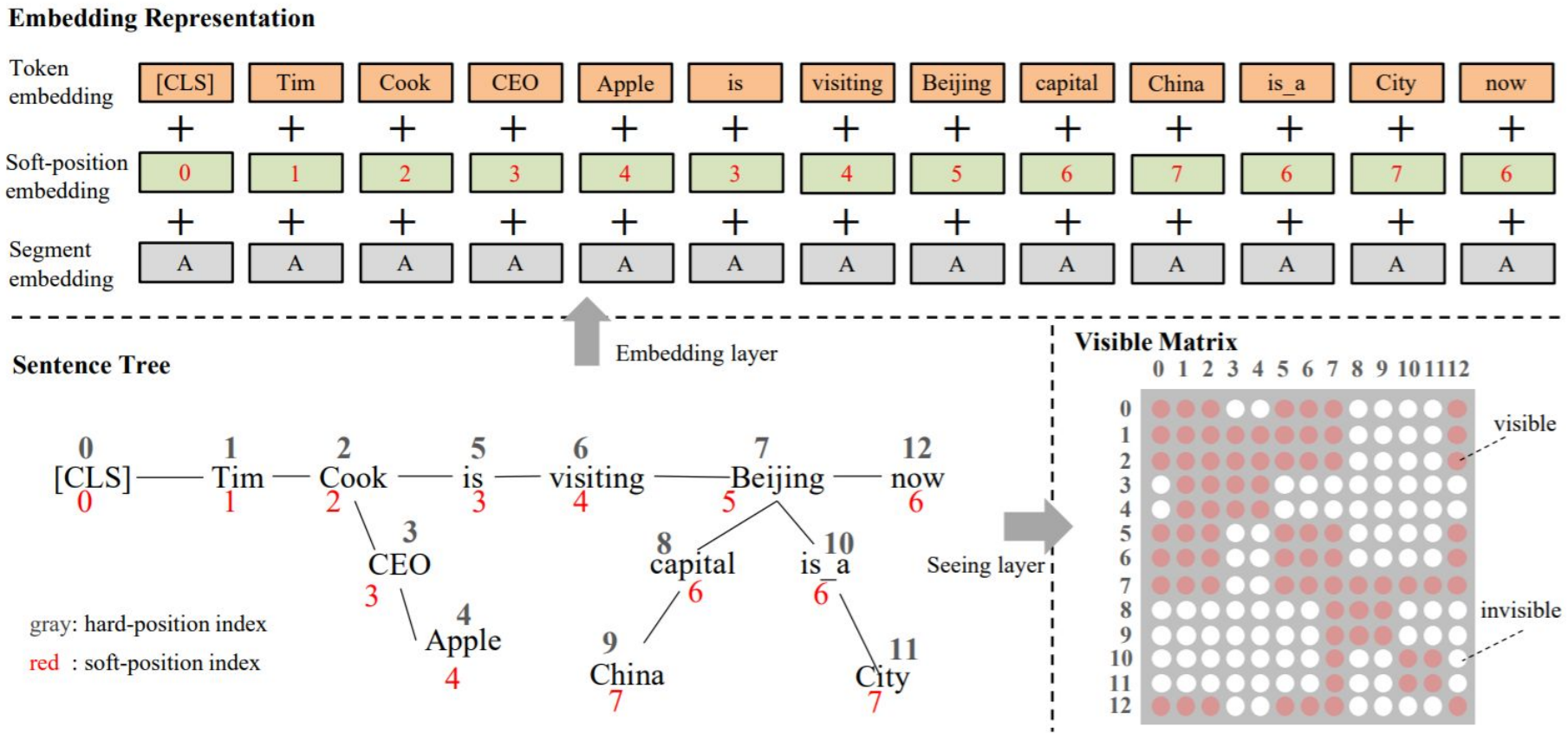
Blocking interaction between nodes in different branches to *avoid knowledge noise*.

Matrix with entries M_{ij}:

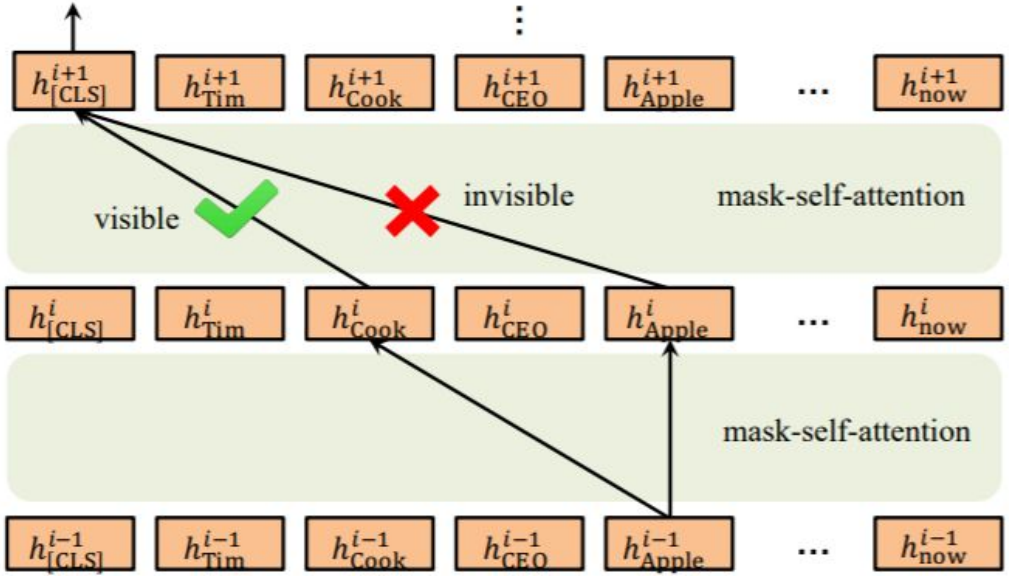
$M_{ij} = 0$ if w_i and w_j on same branch
$M_{ij} = -\infty$ otherwise

(i and j are hard position indices)

K-BERT: Example



K-BERT: Mask-Transformer



Mask Self-Attention

The visible matrix M is summed to the query-key matrix product:

$$Q^{i+1}, K^{i+1}, V^{i+1} = h^i W_q, h^i W_k, h^i W_v,$$

$$S^{i+1} = softmax(\frac{Q^{i+1} {K^{i+1}}^\top + M}{\sqrt{d_k}}),$$

$$h^{i+1} = S^{i+1} V^{i+1},$$

If $M_{jk} = -\infty$ then the attention score $S_{jk} = 0$, then the hidden states h_j and h_k do not affect each other.

Open-domain evaluation: choose your Knowledge Base wisely

HowNet contains **linguistic knowledge** (word-sememe relations).

CN-DBPedia contains open-domain **encyclopedic knowledge**.

In **open-domain evaluation**:

1. KGs have **no significant effect** over baseline **on sentiment analysis** (i.e., Book review, Chnsenticorp, Shopping and Weibo)
2. Language KG better than encyclopedic KG and baseline in semantic similarity tasks (i.e., XNLI and LCQMC)
3. Encyclopedic KG better than language KG and baseline for Q&A and NER tasks (i.e., NLPCC-DBQA and MSRA-NER)

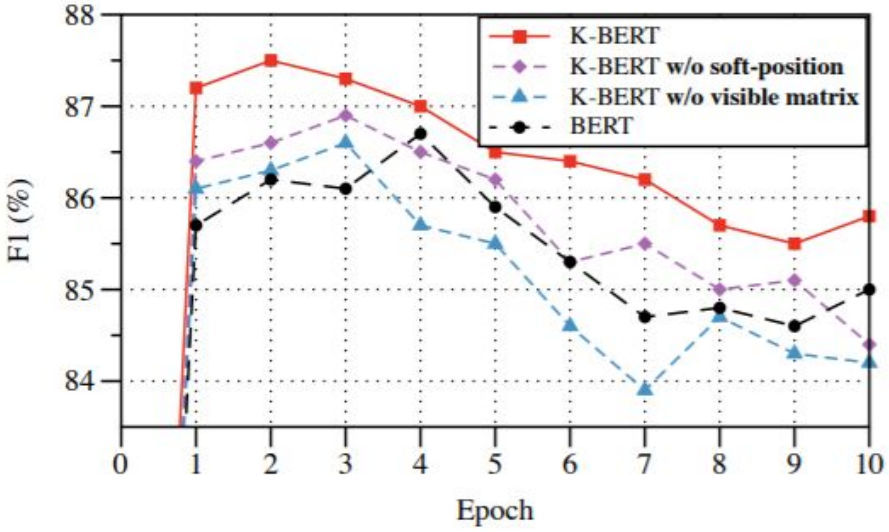
K-BERT: Domain-specific Evaluation

Dedicated, **domain-specific KGs are even better** for domain-specific NER and Q&A.

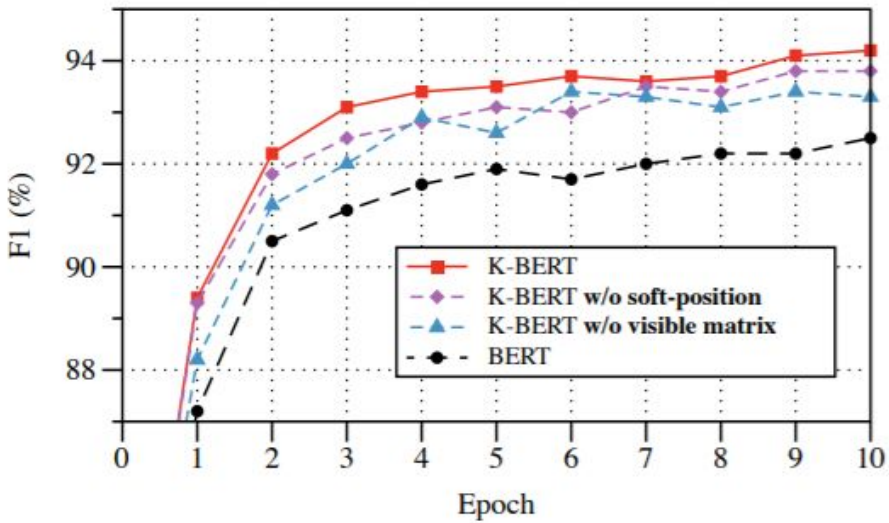
Models\Datasets	Finance_Q&A			Law_Q&A			Finance_NER			Medicine_NER		
	P.	*R.*	*F1*	*P.*	*R.*	*F1*	*P.*	*R.*	*F1*	*P.*	*R.*	*F1*
Pre-trained on WikiZh by Google.												
Google BERT	81.9	86.0	83.9	83.1	90.1	86.4	84.8	87.4	86.1	91.9	93.1	92.5
K-BERT (HowNet)	83.3	84.4	83.9	83.7	91.2	87.3	86.3	89.0	**87.6**	93.2	93.3	93.3
K-BERT (CN-DBpedia)	81.5	88.6	**84.9**	82.1	93.8	**87.5**	86.1	88.7	87.4	93.9	93.8	93.8
K-BERT (MedicalKG)	-	-	-	-	-	-	-	-	-	94.0	94.4	**94.2**
Pre-trained on WikiZh and WebtextZh by us.												
Our BERT	82.1	86.5	84.2	83.2	91.7	87.2	84.9	87.4	86.1	91.8	93.5	92.7
K-BERT (HowNet)	82.8	85.8	84.3	83.0	92.4	87.5	86.3	88.5	87.3	93.5	93.8	93.7
K-BERT (CN-DBpedia)	81.9	87.1	**84.4**	83.1	92.6	**87.6**	86.3	88.6	**87.4**	93.9	94.3	94.1
K-BERT (MedicalKG)	-	-	-	-	-	-	-	-	-	94.1	94.3	**94.2**

K-BERT: Ablation Study

1. Both soft-position embeddings and the visible matrix are crucial.
2. K-BERT w/o visible matrix worse than BERT in Law Q&A: knowledge noise.
3. K-BERT converges faster than BERT (epoch 2 vs epoch 4 in Law Q&A).



(a) Law_Q&A



(b) Medicine_NER

Table of Contents

Takeaway messages 1/3

Specialization is still indispensable for CWEs: the distributional signal distorts lexical relations and does not recover world knowledge entirely.

Three main approaches to CWE specialization: i) Masking strategies. ii) Multi-task objectives. iii) Graph reshaping.

Multi-task objectives include relation prediction (LIBERT), denoising auto-encoding (THU-ERNIE), and un-/weakly supervised tasks (ERNIE 2.0)

Takeaway messages 2/3

Most specialization methods operate during the pre-training phase, with the exceptions of K-BERT (refinement phase) and THU-ERNIE (both).

Linguistic knowledge benefits acceptability, NLI, and text similarity tasks (cf LIBERT, ERNIE 2.0), while world knowledge excels in NER and Q&A tasks (cf ERNIE 2.0), entity and relation classification (cf THU-ERNIE).

The specialization method matters: e.g. GLUE performance higher than baseline for continuous learning (ERNIE 2.0), mildly higher for masking (Baidu-ERNIE), and comparable for denoising entity AE (THU-ERNIE).

Takeaway messages 3/3

Only reshaping models text-graph interactions. Other methods have to discard relational information or links between texts and graphs.

Specialization is beneficial for any data size (cf LIBERT), but especially for few-shot learning and learning on noisy labels (cf THU-ERNIE).

Specialization with domain-specific KGs outperforms general-domain KGs for in-domain applications.

Main Bibliography

- Lauscher, Anne, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. "**Informing unsupervised pretraining with external linguistic knowledge**." arXiv preprint arXiv:1909.02339 (2019).
- Liu, Weijie, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. "**K-BERT: Enabling Language Representation with Knowledge Graph**." arXiv preprint arXiv:1909.07606 (2019).
- Petroni, Fabio, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. "**Language Models as Knowledge Bases?**." arXiv preprint arXiv:1909.01066 (2019).
- Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. "**ERNIE: Enhanced Representation through Knowledge Integration**." arXiv preprint arXiv:1904.09223 (2019).
- Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. "**ERNIE 2.0: A continual pre-training framework for language understanding.**" arXiv preprint arXiv:1907.12412 (2019).
- Zhang, Zhengyan, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. "**ERNIE: Enhanced Language Representation with Informative Entities**." arXiv preprint arXiv:1905.07129 (2019).

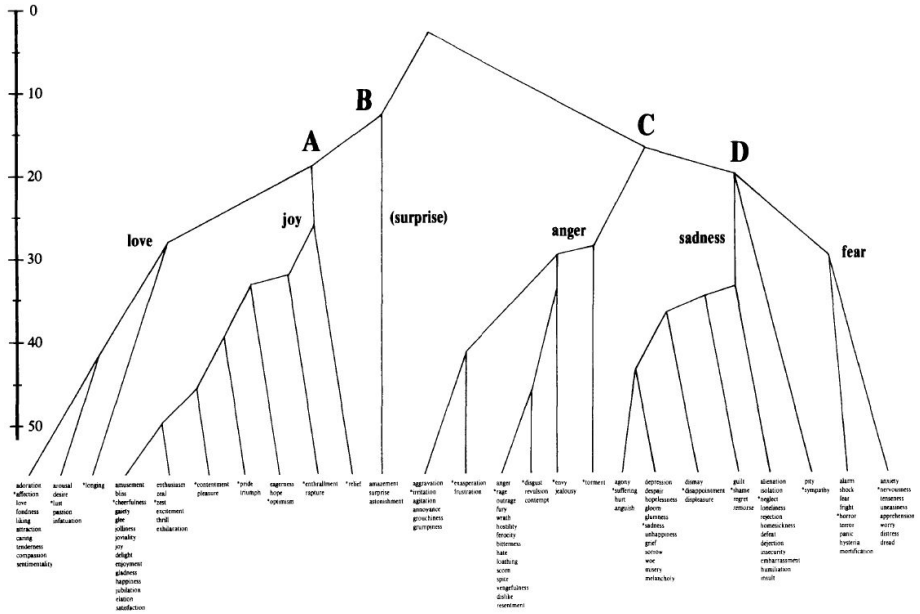
Final Thoughts

Open challenges: Lexical fields

Injected knowledge usually takes the form of pairwise constraints.

However, lexical semantics is based on the notion of **lexical field**, which entails multi-way relations among a set of words. E.g. the area of emotions (Shaver et al. 1987)

Specializing for lexical fields would require to **fine-tune the local topology of semantic sub-spaces**.



MDS on post-specialized embeddings

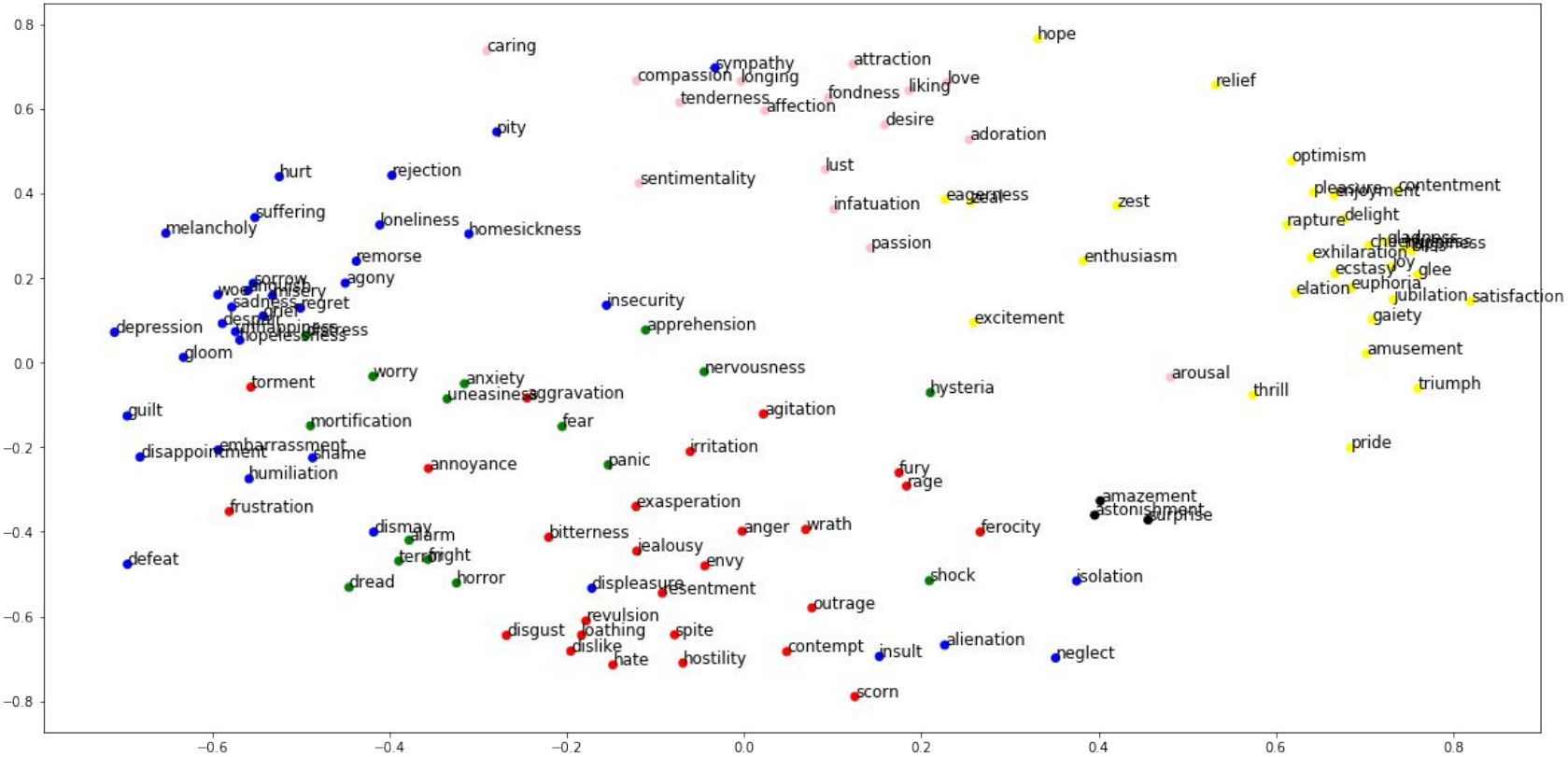


Image courtesy of David Russell Richie

Open challenges: Reverse Specialization

After cross-lingual specialization, the target specialized space of a resource-poor language could guide the automatic creation of (noisy) word-level lexical resources.

Target constraint induction (Ponti et al. 2019) directly leads to such resource. How to extend this idea to multiple lexical relations?

Multilingual vector alignment can be adapted for this purpose by “discretizing” the continuous relations of closeness.

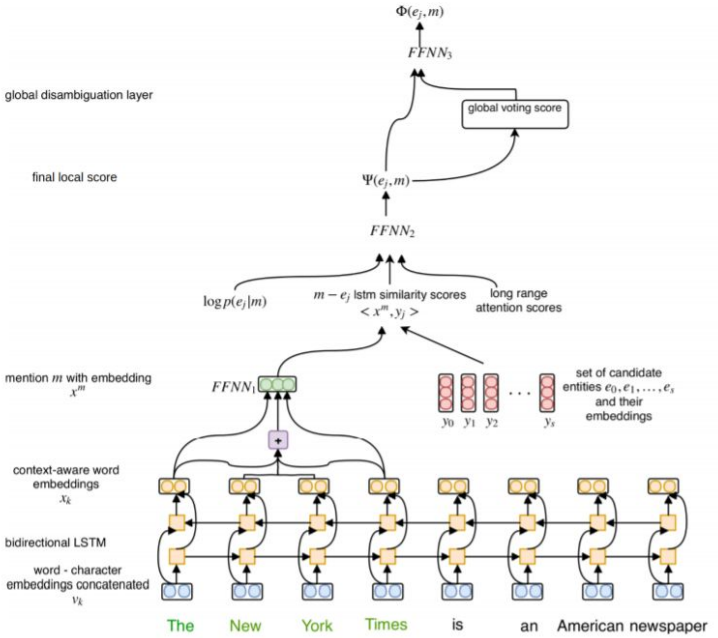
Open challenges: Weakly supervised Entity Linking

Specialization methods for CWEs **assume the existence of NER taggers**.

For most languages, they are **unavailable or inaccurate.**

Entity Linking could be achieved as an auxiliary pre-training objective.

Wikipedia dumps are labelled for links.



End-to-end neural entity linking (Kolitsas et al. 2018)

Takeaway messages 1/3

The goal of integrating a) distributed representations with b) structured knowledge is mitigating their respective limitations: a) conflates different relations, while b) has low coverage (of words and languages).

Relations have different natures: e.g. symmetric vs directional, graded or not. Their specialization demands different methods.

Pros and cons of methods: joint learning affects all the words in the vocabulary. Post-processing shows better performances, is not tied to specific embedding models, and needs no retraining.

Takeaway messages 2/3

Limited vocabulary coverage calls for post-specialization or explicit specialization.

Linguistic specialization has been repeatedly proven to boost performances in Dialog State Tracking, Lexical Simplification, and Text Similarity.

Specialization can be transferred across languages via multilingually aligned semantic spaces, or by inducing target constraints.

Takeaway messages 3/3

The specialization framework has broad applicability: bio-NLP, debiasing, abusive language detection, fact checking, cognitive studies….

Not all methods model full triples (word-relation-word). Some focus on single-relation constraints, attract and repel relations, or unbounded relations (functional extensions).

Specialization is beneficial to both static and contextualized WEs. But there is still a lot to be explored, especially about the latter.